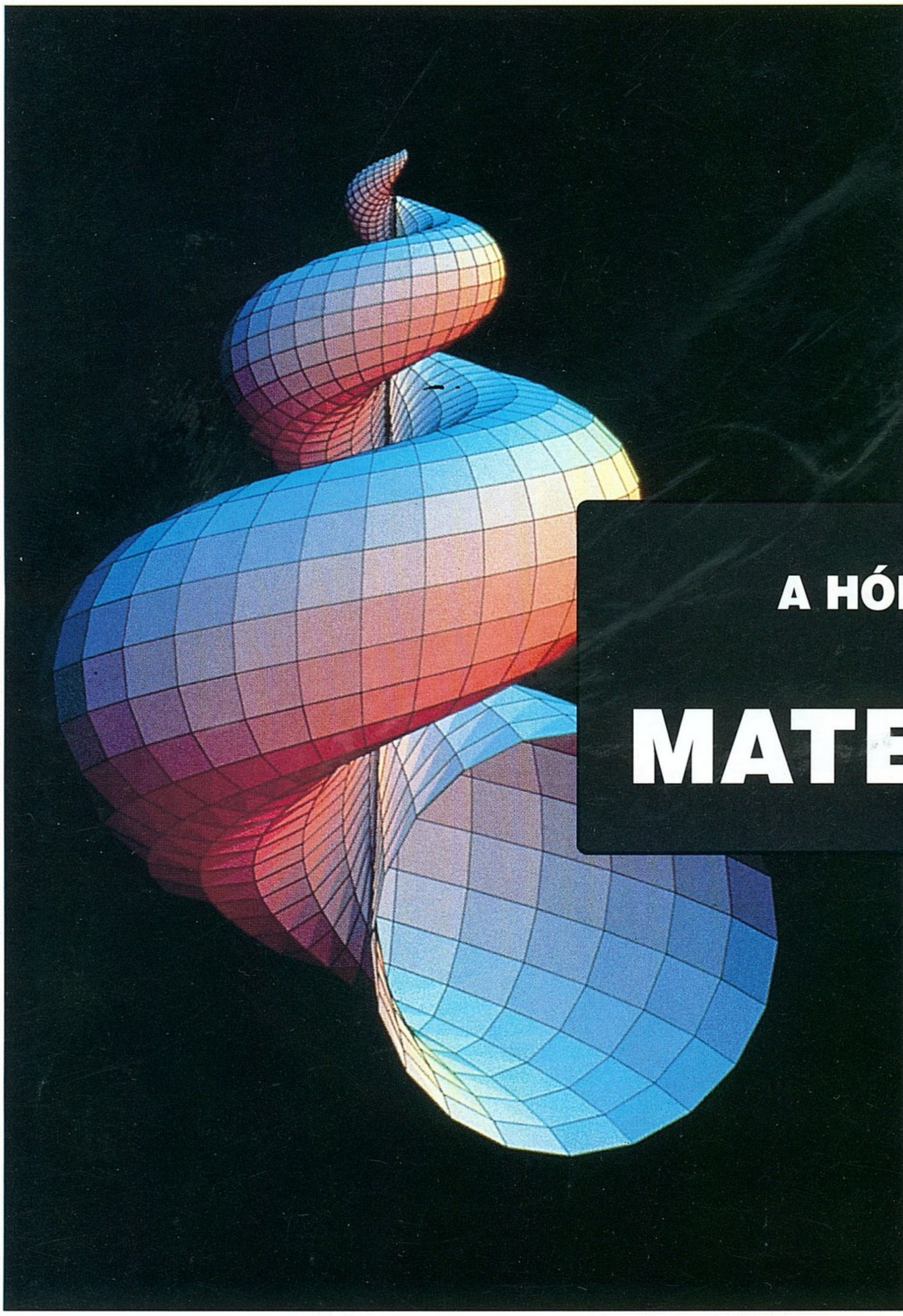
XV. ÉVF. 9. SZÁM, 1997. SZEPTEMBER ÁRA: 396 FT

ÚJ ALAPLAP

SZÁMÍTÁSTECHNIKAI FOLYÓIRAT LEMEZMELLÉKLETTEL



4DOS 6.0
DOS Doktor-tipp
Ismét a szabályosságról
Szimbolikus matematika
Makróvírus-ellenőr
PC Rébusz 97/9

A HÓNAP TÉMÁJA:

MATEKIZMUS

Újra DOS Doktor

Visual DataFlex

Sakkranglista

Irány a CD!

Az Office 97 és a vírusok

Jönnek a nanoszámítógépek?



INFORMÁCIÓKÉRÉS: 0925 ▲

ÚJ ALAPLAP

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató számítástechnikai folyóirat

Megjelenik havonta, mágneslemez melléklettel

Főszerkesztő:
Faklen Pál

Főszerkesztő-helyettes:
Varga János

Szerkesztő:
Jakab Ágnes

Szerkesztőség és kiadó:
1539 Budapest
VI., Dózsa György út 84/b
Telefon: 322-4417, 322-5238
Fax: 214-9492
E-mail: alaplap@mail.datanet.hu

Felelős kiadó:
Faklen Pál

Terjesztés:
Megyes Zsuzsanna

Hirdetésszervezés:
Árvai Katalin, Bogácsi Mária, Galyasi Hedvig, Pap Katalin

Külföldi hirdetések:
PubliCity
Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 156-1182 Fax: 175-3539

A kiadó a hirdetések tartalmáért és a nyomdakészen kapott hirdetések formájáért (és helyesírásáért) nem vállal felelősséget

Példányszámadatok hitelesítése:
Magyar Terjesztésellenőrző Szövetség
MATESZ
Ez a szám
10 000 példányban jelent meg

Nyomtatás:
Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:
Somogyi Tibor ügyvezető igazgató

Terjeszti:
A Magyar Posta Rt, a Nemzeti Hírlapkereskedelmi Rt, a Hírker Rt, a Kiadói Lapterjesztő Kft, számos számítástechnikai szaküzlet és más alternatív terjesztő

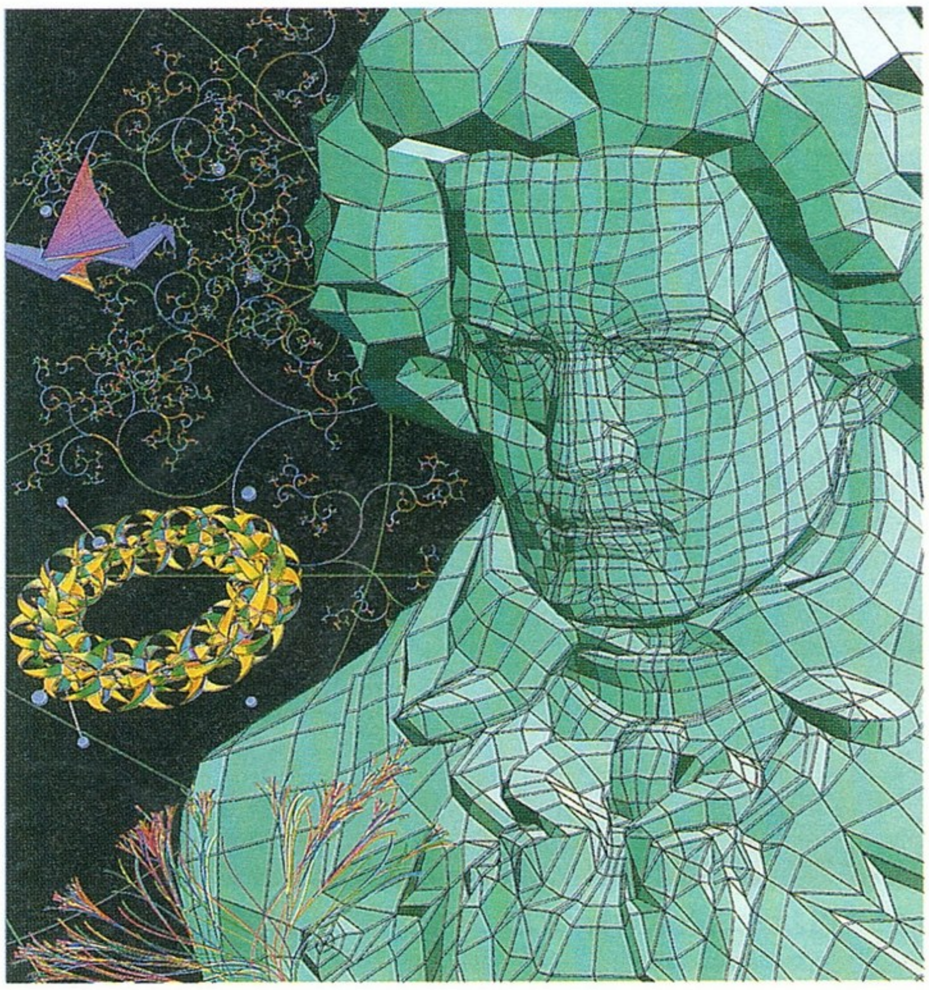
Előfizethető a kiadónál:
Új Alaplap Kiadói Kft,
1539 Budapest, Pf. 571
Bankszámlaszám:
OTP 11706016-20788599
Eladási ár: 396 Ft
Évi előfizetési díj: 4356 Ft

Évi külföldi előfizetés díja:
4356 Ft + postázási költség

HU ISSN 1217-7598

TARTALOM

A gondolkodás élvezete

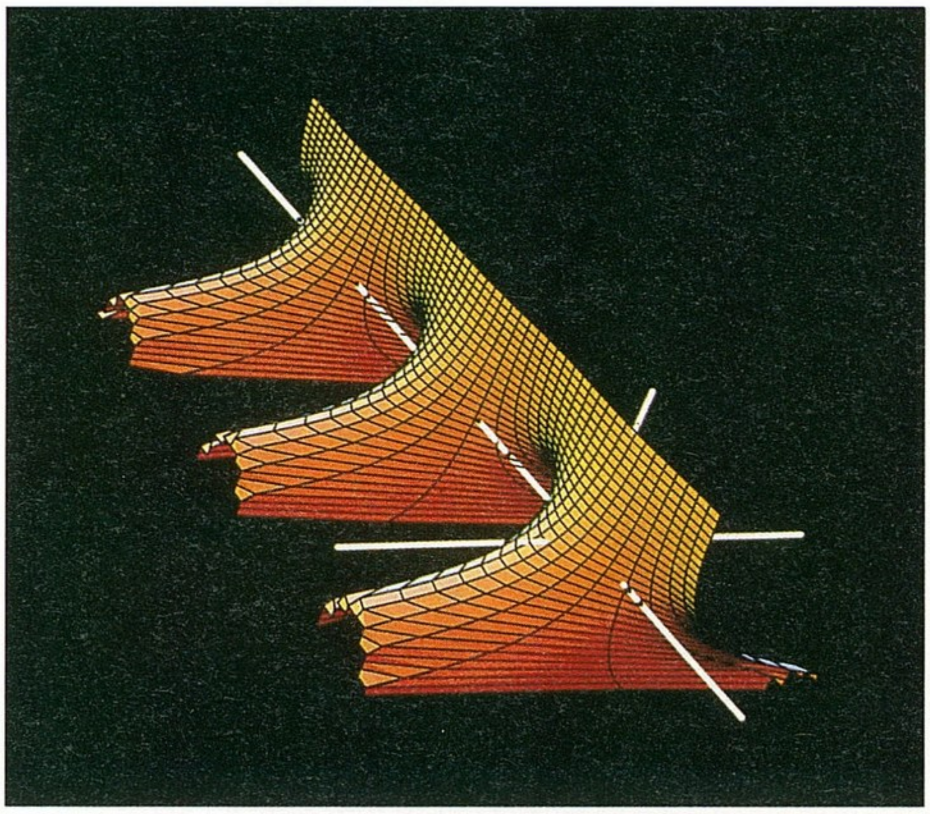


A matematikát lehet szeretni és lehet utálni, de nélküle nehéz lenne megélni. Ma már szinte egyetlen tudomány sem tudja mellőzni a matematikai eszközöket, így az irodalmi, orvosi, történeti, szociológiai szakcikkeknek is elmaradhatatlan részei a statisztikák, a matematikai eszközökkel végzett elemzések, bizonyítások.

Azt még mindenki tudomásul vette, hogy a természettudományokban mindennapos vendég a matematika, de azzal nehezebb volt megbarátkozni, hogy például századunk nyelvészetére is rányomta bélyegét a generatív grammatika, azaz a formális nyelvek elmélete, vagy hogy az alternatív gyógymódok közül némelyik a fuzzylogikára és a káoszelméletre igyekszik eljárását alapozni.

A számítástechnika is matematikai alapokra épült. A tudományág bölcsőjénél a matematikából ismert tudósok tevékenykedtek, és azóta is sok matematikus dédelgette eme szakterületet. Ennek ellenére igen sok komoly matematikus máig is lenézi a számítástechnikát, nem tartja tudománynak, csak egy eszköznek, és még eszközként sem becsüli sokra.

Kénytelen-kelletlen azonban a matematikusok is elkezdték a gépeket használni. Meglepő módon az olyannyira elmélet felé hajló matematikusok körében a számítógépesítés egy nagyon is praktikus területen nyerte meg első csatáját. Sokáig ugyanis elég idegőrlő munka volt egy matematikai képleteket tartalmazó irományt kiadni. A kéziratot valakivel le kellett gépeltetni, s mivel a titkárnők nem kaptak matematikai képzést, nem csoda, ha egy felső indexben szereplő mennyiség alsó indexét rossz helyre írták, mert nekik ez így is, úgy is kínaiul volt. Ezért gyakran egész oldalakat kellett újragépeltetni, majd újra átnézni — mindaddig, amíg az végre jó nem lett. A matematikus ráadásul nem átall a különféle indexek mellett mindenféle törteket, integrálokat, sőt görög, héber vagy akár gót betűket is használni. Mindezekkel megbirkózó írógép nem is sok akad, tehát a már legépelt szövegbe rendszerint kézzel kellett beírni a képletek nagy részét.



(Hadd szenvedjen vele azután már a nyomdai szedő.) A szövegszerkesztő programok megjelenésével mindez alaposan megváltozott. Az első időszakban a diákoknak gyakran zsebpénzkiegészítést jelentett a matematikai szövegek „sajtó alá rendezése". Később már a kutatók, tanárok maguk is be tudták írni szövegeiket, az új nemzedék pedig egyenesen beleszületett a számítógépek aktív használatába.

Ebben az összeállításban megpróbálunk felvillantani néhány érdekességet a matematika számítástechnikai kapcsolódásairól, majd néhány konkrét programon keresztül igyekszünk mindenkihez közelebb vinni a számítógép matematikai alkalmazását, végül pedig felvetünk néhány provokatív gondolatot — mindezt nemcsak matematikusoknak vagy a matematika iránt vonzódóknak szánva, hanem mindazoknak, aki számára a gondolkodás élvezet.

Meddig számtan, honnan matek?

Géppel számolni

Az emberiség régi szokása, hogy megszámolja a körülötte található dolgokat, s így valamilyen értelemben birtokba veszi, megszerzi azokat. Számolást segítő segédeszközöket igen hamar elkezdtünk készíteni. Mindenki számára ismerős az abakusz, a ma már játéknak minősülő, fakeretes, drótos számológépek őse. Azután megjelentek a mechanikus összeadógépek, amelyek később az analóg számítógépekhez vezettek el.

A technika fejlődésével a piacokra és az utcai (fekete) pénzváltókhoz is eljutott a négyműveletes számológép, amely kezdetben tiltott, mára megtűrt szereplője lett az iskolai matematikaóráknak. A lapos kis szerkezet pillanatok alatt elvégzi azokat a szorzásokat, osztásokat, melyekkel mi még jócskán megkínlódtunk.

Tábla helyett nyomógomb

Lassanként eltűnik a „négyjegyű" is, az a táblázatgyűjtemény, amelyből a szögfüggvények értékeit, a logaritmusokat, a négyzet- és köbgyököket kereshettük elő. Nekem hivatalosan már nem oktatták, hogyan vonhatunk négyzetgyököt egy számból, pedig az korábban a tananyag része volt. Ugyan mi minden fog még eltűnni napjaink tananyagából az elkövetkező évtizedben? Mennyire kell besulykolni olyan műveleteket, amelyek elvégzésére ott van a zsebszámoló, vagy tényleg elegendő csak a számolási elvet ismerni?

Vannak persze olyan számolási feladatok is, amelyekre ezek a zsebszámológépek jelenleg még nem képesek. Egyenletek, egyenletrendszerek megoldása, egyszerűbb alakra hozása ma még a mi feladatunk, noha ezek is könnyen algoritmizálható feladatok, tehát készíthető rájuk program. Az olyan programokat, amelyek nemcsak az eredmény számszerű megoldására képesek, hanem paraméterekkel is hajlandók dolgozni, szimbolikus matematikai vagy komputeralgebrai rendszereknek hívjuk. Ezek általában a következőkre képesek: képletek egyszerűsítése, adott változó szerinti kifejtése, szorzattá alakítása, gyökeinek, deriváltjainak, integráljainak meghatározása, egyenletrendszerek megoldása. A kiszámolt eredményt megjeleníthetik grafikusan, az adatokat, képleteket kimenthetik lemezre, és onnan vissza is tölthetik.

A komolyabb rendszerek rendelkeznek valamilyen programnyelvvel, és ennek révén az esetleg még kihagyott függvényeket vagy más objektumokat pótolhatjuk. Vannak nyitott rendszerek, amelyekhez megszokott programnyelvünkön is készíthetünk kiegészítéseket, így az egyszer már elkészített eljárásokat, függvényeket később is használhatjuk.

Mivel e rendszereket remekül használhatják vegyészek, fizikusok, biológusok, közgazdászok, statisztikusok és mindenki, aki képletekkel dolgozik, nem csodálkozhatunk azon, hogy ez is szoftverpiaci szelet lett, és sorra készülnek az egymással rivalizáló programok. Szerencsére akad néhány ingyenes verzió is, bár ezek inkább csak demóváltozatnak tekinthetők, mert a komolyabb (nehezen programozható) funkciók hiányoznak belőlük.

Derive — mindenkinek

Az első programrendszer, mellyel találkoztam, a Derive volt. DOS alatt fut, és nincsenek nagy igényei a géppel szemben. A középiskolában felmerülő ismereteket tudja, így egyre több középiskolai tanárral próbálják megismertetni, hogy ők is használják fel azt az oktatásban.

Emlékszem, hogy a nyolcvanas évek elején azzal próbálták eladni az Commodore 64-et és a hasonló kategóriájú gépeket, hogy azok segítenek a gyerek házi feladatát elkészíteni. Én azonban egy elvetemült nyelvtani házi feladaton kívül ilyesmire nem tudtam használni gépemet a középiskolában. Egyetemista koromban viszont egyszer egy ilyen programrendszerrel sikerült határidőre megoldani a kiadott feladatokat.

Úgy vélem, hogy bizonyos alapozásra mindenképpen szükség van. Ha a helyzet úgy adódik, mindenki legyen képes számológép nélkül is kiszámolni, mibe kerül két kiló krumpli meg fél kiló hagyma. Ha viszont az iskola megengedheti, hogy néha gép mellé ültesse diákjait, akkor hadd próbálgathassák azt is, hogyan függ az $y=(x-u)^2+v$ függvény alakja az u-tól és a v-től. Igaz, ilyet mi is csináltunk, de kézzel rajzoltuk fel a függvényeket, és fejben számoltuk ki a függvényértékeket az adott pontokhoz. Vannak nagyon érdekes függvények, de ezeket papíron kiszámolni és kézzel felrajzolni bizony nem megy gyorsan. Volt persze olyan egyetemista, aki logarléccel számolt ki függvényeket, és milliméterpapíron ábrázolgatta őket. (Nem tudom, hogy esténként hány függvénnyel készülhetett el, de sokkal biztosan nem.)

Program a lemezmellékleten

A szimbolikus programcsomagok nemcsak rajzolásra valók, de talán ezek a leglátványosabbak, ezért érdemes velük kezdeni az ismerkedést. Hogy eddig eljussunk, kezdjünk el egy gyors tanfolyamot. Azért, hogy ne a levegőbe beszéljek, és az elmondottakat ki lehessen próbálni, a lemezmellékletre feltettünk egy egyszerűbb rendszert. Mivel a program szerzője ebből a programból akar megélni, ez a verzió bizonyos korlátozásokat tartalmaz, de talán még így is sikerül ráérezni az ilyen programok ízére.

Azért esett a választásom éppen erre, mert nem igényel semmilyen különleges hardvert, mégis viszonylag sokat tud. Az ablakos, menüs felület könnyen kezelhetővé teszi a rendszert a kezdőknek is. A menüsort az F10 billentyű lenyomásával érhetjük el, és ott a kurzorbillentyűkkel mozoghatunk. Az INPUT ablakban fogalmazzuk meg kérdéseinket, és a futtatás (RUN) után az OUTPUT ablakban kapjuk meg rá a választ. Kezdetben érdemes gyakran használni a HELP menüpontot és az SM kiterjesztésű példafájlokat. A program lehetővé teszi, hogy az egyes ablakok tartalmából átmásolhassunk egy másikba, ez segíthet megismerni a programot, mert a helpben szereplő példákat így könnyedén kipróbálhatjuk. A program szövegszerkesztője a WordStar kiosztását követi, de ha valahol elakadnánk, üssük le az F1 billentyűt.

A rendszer programozási szerkezeteket is tartalmaz, hogy egyszerűbben megfogalmazhassuk igényeinket, így például a feltételes szerkezeteket, cik-

lusokat. További segítséget jelent, hogy az input és az output fájlokra tördelhető, és megszabhatjuk, hogy mit honnan olvasson be, milyen eredményeket hova írjon. Ezzel ezt a programot könnyedén beilleszthetjük meglévő környezetünkbe. Például a program sorra beolvassa az inputfájlokat, és elkészíti az eredményfájlokat, majd a szövegszerkesztő mindezekből összeállítja a kész cikket, s nem nekünk kell darabolni, vagdalni, majd összemásolni a megfelelő részeket. A program saját igényeinkhez történő igazítására többtucatnyi beállítást használhatunk. Ezek közül egyesek az inputra, számolásra, míg mások a kimenetre lesznek hatással.

Precíz és „rigolyás"

Ha csak néhanapján vesszük elő ezt a programot, akkor nincs szükség megjegyzésekre. Más esetben viszont hasznos néhány szóval leírni, mit is akarunk csinálni, hogy legközelebb pillanatok alatt felvehessük elgondolásunk fonalát.

A számológépekhez szokottak első meglepetése az lesz, hogy az 1/3-ból nem lesz azonnal 0,333333333. Ez azért van, mert ezek a rendszerek mindaddig nem kerekítenek, amíg tudnak pontos értékekkel számolni. Külön kérésre persze hajlandók számszerű értékkel válaszolni. Meglepő az is, hogy milyen „rigolyás" a program. Ez azért van, mert ha nem állítunk mást, akkor minden

A Pisot number

$$\left(\sqrt[3]{\frac{2}{27+3\sqrt{69}}}+\frac{1}{3}\sqrt[3]{\frac{27+3\sqrt{69}}{2}}\right)^{30000}=$$

504316596065665493215059469242757481077610274680893503167844467712875718226486210610398076581934068275385782540821330947870
035477787176694685987718254576565515786375180403084108087720197360263045377599299252600787464343711547834066265665424226316
713797047087646002867659572088255372994141399119295425729390551598734162316493172898778929854821816913956381635362412821599
451025319364903651110221515233814364527904893091337135424027258899192161986823235528295581059873200107875550497563219302434
498746973208148942951387505081508954467615093825332988410762937088778366046553068685772855386645950482646814439301999016574
337736406592225567911709051797128549077311912778888144108736659703862174018641106917574995202365941964030217895956126590152
078242574779075678185904725211878343979501048953268614290340265384740790514648773550021687547584525222504190436087068107827
922710445722736236879669324173378416116695919841030467768296491940904387052300273200377023748044343103233169109426144114438
029834875192019905956279051188804975026500268288352191389685849628968612357880766725385147857934444955391441537519437285121
796122315842862423189949212117527981448636085804805657512086481790698235946554679103334557030614430688918237659355818871823
118397751118724092116272862912671381160865803740893007151687615872534112762296886660521036261139972821525679422459991243185
253122720544532313520622663794272436418750735995618191121806899179577732679626160875234377580466842238136144609068158859423
168444990563130856914398161104641435976464505827310532042357277805504151076128744133166097040034356155594355699310372567007
362846011243924380051721179466481515865177147916094265318673753576167289982789156765819918501461470394696987900055057821804
367427737785530796529257310070440503552865770656588911341336781719293479609751349965199241672156885820419323697826495675216
785846761301665051497508373554704845376808917106579182046296637409148513289873712928793514345917581372183082138176837444493
669698475749155502722721819149673730689452542949330172497476348802878923380972880989819941626981662787664453931670154635431
610660994478782022049824681885420029046846763098262298893067982085873006027223547673545004480573914810092491101933870588744
990654481144456437881423922057266818027803279734701366817533469290959062710889715842108254898313134138767612788549048657824
912624883639420318879265232485613570477399709046710448077813242383525775842418033574517057365270541979506919519088585109853
839062549850746730941961116578083869871331034408636054845107081922257029648252724270305699507491876725741414282227189033471
727764395434254185950990747290625952762962184681171936790178602417989512006492961330651781076337449499729308205550243207661
846458602761611981567677373636018999149195717505734613349483793082747826653826109388871646107637707669952322121063841012111
950272802819214732121339193913851711740180545436004534753223529995802192329202767788406529942228608625210506805680779347684
445472832022947989505073864225065042453778536680082426933681881662317694955672636170867065177268568641882560696136399378938
092363061534206649910677121667830726528065065115430787822086187067816894109743170406015960678013961777060204838325510730464
182561678677866776664114588829507728845970964346546857260085067375205588321834061968066477136954754650582659722182787527038
013350885360773847158970095196369406644986845955931896822762457057294441289666846719625035790435931009430361335058057283719
525424750533492243401220689154316788170873900888499149058827913507680455068280339445423082211488881873764963543994136415115
21268300533409275323047144881568429552172625312395819678941865586820107952918459296302076757812 53.0000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000001308948114465174795123603548945492907...

változót komplexnek tekint. Erre általában nincs szükség, többnyire megelégszünk azzal, hogy a számaink pozitívak. Ezt azonban érdemes a program tudomására hozni az assume() paranccsal, mert utána már nem lesznek annyira riasztóak az eredmények.

A program ismer néhány matematikai konstanst, így például a pi-t, vagy a természetes logaritmus alapját, az e-t. A különféle mennyiségek tárolására a változókat használhatjuk, és ha másként nem rendelkezünk, lényeges, hogy kisbetűvel vagy nagybetűvel írtuk őket. A változók értékadására a Pascalból ismert := szolgál. Mivel igen sok változót, függvényt használhatunk, ezért közkedvelt játék megadni a különféle átváltási értékeket, és ezután oda-vissza váltogatni a különféle mértékegységek között. A változókat a számoláson kívül egyébre is lehet használni, így például feltölthetjük különféle adatokkal, és azokat könnyedén visszakereshetjük, tehát például címjegyzékeket is készíthetünk. A túl sok definíció sok memóriát foglalhat, ezért időnként a felesleges változókat töröljük a memóriából a clear() paranccsal! Definiálhatunk függvényeket is, erre három egymástól kissé eltérő módszer is szolgál. Lássuk talán a legegyszerűbbet:

f(x_,y_) := x^2 – x * y + y^2,

vagy másként írva:

$f(\underline{x},\underline{y}) := x^2 - x \times y + y^2$.

Függvényeink definiálásakor kihasználhatjuk a program képességeit, így akár feltételeket vagy ciklusokat is szerepeltethetünk a definícióban. Figyeljünk arra, hogy definiáláskor a függvény argumentumai mögött szerepeljenek az aláhúzás karakterek! Az általunk definiált függvények kiegészítik a rendszerben már meglévőeket. Ez utóbbiakat két csoportba oszthatjuk, belső és külső függvényekre. A belsők „bele lettek égetve" a programba, azokat nem változtathatjuk meg. A külsők az LI kiterjesztésű fájlokban találhatók meg, és csak akkor töltődnek be a memóriába, ha szükség van rájuk. Ha úgy kívánjuk, akár ezeket a fájlokat is megváltoztathatjuk, törölhetünk belőlük vagy kiegészíthetjük őket.

Sok rontott megoldásnak a hibás visszahelyettesítés az oka. Ha bonyolultabb képletbe kell bonyolult értéket behelyettesíteni, akkor ezen nem is nagyon lehet csodálkozni. A számítógép viszont könnyedén megbirkózik ilyen feladatokkal:

subs(– x^2 + 25 * x – 40, x = y/3 – 2).

Képlet helyett használhatjuk a „last" szócskát is, mert ez az előző művelet eredményét jelenti.

Gyakran nem elégszünk meg az eredmény formájával, olykor fel akarjuk bontani a zárójeleket, máskor meg az egészből szorzatot szeretnénk csinálni. Az első feladat az egyszerűbb, erre az expand() szolgál. A szorzattá alakítás általában nem megy olyan könnyedén. Lényegében a polinom gyökeit kell megtalálni, amelyekből a szorzat pillanatok alatt elkészíthető. A gyökök meghatározására ötöd- vagy magasabb fokú polinom esetén nincs sok remény (már Galoa is megmondta), az alacsonyabb fokú esetekre meg ebből a programból hiányzik a szükséges megoldóképlet.

Deriválás, integrálás

Nagy csinnadrattával harangozza be a szerző, hogy a program tanulni is képes. Persze mindössze arról van szó, hogy például ha megadjuk egy függvény integráltját, akkor az abból egyszerűen képezhető függvények integráltját a program már önállóan is meghatározza. Ez nem olyan nagy durranás, de azért esetenként jól jön.

A program elvileg képes deriválni, és akár egyoldali deriváltakat is meghatároz. Függvényeinknek megkereshetjük a határértékeit, akár a végtelenben is. Ha ez nem vezetne eredményre, megpróbálhatjuk a szimbolikus módszer helyett a numerikusat. Az integrálás már kemény dió, sokak szerint itt kezdődik a matematika, ami addig van, az csak számtan. Egy kis segítséggel a program igen sok függényt képes integrálni, de csodákra azért ne számítsunk. Ha nincs szerencsénk, itt is választhatjuk a numerikus integrálást. Hasonlóak érvényesek a sorokra, sorozatokra.

Többek között úgy próbálja megkötni a kezünket a program, hogy az egyenletek gyökeit megadó parancsot csak a fizető felhasználó használhatja. Ha nem várjuk el, hogy a szánkba repüljön a sült galamb, hanem teszünk is valamit, akkor azonban a gyökök meghatározásában így is jó segédeszköz lesz. Változóval hivatkozhatunk egész egyenletekre, majd ezek megfelelő többszöröseit összeadva kiejthetünk egyes tagokat. Ezt egyszerűsítve, és az eredeti egyenletbe visszahelyettesítve már a kívánt eredményekhez jutunk. Ugyanazt kell csinálni, mint papírral és ceruzával, viszont nem bonyolódunk bele a részletekbe, és ez talán még az oktatásban sem hátrány. A solve( ) használatával persze mindez jóval egyszerűbb lenne, de valamit valamiért. A teljes verzióval egyszerűbb differenciálegyenletek is megoldhatók.

Az eredményt „eladni"

Használhatunk a programban tömböket, azaz vektorokat és mátrixokat, valamint listákat. A listák azért jók, mert ugyanazt a műveletet egyszerre akár több száz értéken is végrehajthatjuk. A mátrixokra és vektorokra csak a legalapvetőbb műveletek vannak definiálva, tehát ha ilyen mennyiségekkel akarunk sokat számolni, akkor nem árt saját definíciókat gyártani. Hasonlítanak a listákhoz a táblázatok is, ám ezeket arra érdemes használni, hogy más programoknak átadjuk egy függvény értékeit, például a gnuplot-nak a kirajzoláshoz. Persze a programból nem kell kilépni ahhoz, hogy egy függvény alakját megnézhessük, mert ebben is található függvényábrázolás, sőt egyéb rajzolás is, bár nem tudom, hogy ez utóbbi normálisan mire használható.

A program oktatásakor a számonkérésnek a feladatmegoldáson túl azt is mindig tartalmaznia kell(ene), hogy a gép által kiszámolt értékeket szövegszerkesztővel is megfelelő módon kommentálják, magyarázzák. Feladatot megoldani majdnem mindenki tud, de én is csak az egyetem vége felé jöttem rá arra, hogy egy matematikai anyag megfelelő dokumentálása nem is olyan egyszerű. Aki korábban találkozik ilyen feladatokkal, az később könnyebben megbirkózhat velük. Az pedig lassan már a matematikára is igaz lesz, hogy nemcsak az eredmény számít, hanem annak találása is. Ettől (is) függ, hogyan tudjuk eladni.

Aszalós László



Wu és Chou nyomdokain

Tételbizonyítási lehetőségek

Az idők során felmerült az igény, hogy a matematikai állítások valódiságát valamilyen úton-módon mechanikusan ellenőrizzék. Tehát legyen olyan algoritmus, amely egy speciális módon megadott (matematikai logikai eszközökkel megfogalmazott) állításra megadja, hogy az tétel-e vagy sem. Már a század első harmadában kiderült, hogy ilyen általános algoritmus nincs. Viszont...

Vannak olyan elméletek, amelyekben minden állításról megmondható, hogy az igaz-e vagy sem. Ezek közül sokak számára a geometria lehet egyedül érdekes. Tarski még a század első felében megadott egy algoritmust, amellyel elvileg minden kérdés eldönthető. Itt a matematikai logika nyelvén felírt állításból fokozatosan eltüntetjük a kvantorokat (a *minden,* illetve *van olyan* állításokat).

Sokváltozós polinomokkal

A szemfülesebb olvasók biztos felfigyeltek az „elvileg" szóra. Az a probléma, hogy ez az algoritmus elemi feladatok eseten is igen sok lépésből áll, és nagyon sokat is kell vele számolni, tehát a gyakorlatban nem alkalmazható. Az előző két évtizedben viszont újabb lendületet kapott az automatikus geometriai tételbizonyítás, többek között Wu kínai matematikusnak köszönhetően.

Az ő módszere azon alapul, hogy koordinátageometriai ismereteket felhasználva, sokváltozós polinomok formájában fogalmazzuk meg a feltételeket és a bizonyítandó összefüggést, majd ezek után pontosan meghatározott sorrendben kiszámítjuk az eredeti és az újabban megkapott polinomok közötti osztások maradékát. Ha ez a maradék nulla, akkor a tétel igaz. (Ez csak a felszín, valójában az eredeti formulákból a változókat próbáljuk sorra eltüntetni.) Elmondva ez így igen egyszerű, de rövidebb polinomok osztásakor is kaphatunk hosszú maradékot. Egy érdekesebb feladat során a maradékok összhossza meghaladta a 90 Kbájtot, pedig azok a legtömörebb formában lettek leírva.

Ez a módszer azért nem tetszetős, mert egy olyan gépre emlékeztet, amelybe felül bedobunk egy állítást, és alul kijön egy szó: TÉTEL, vagy pedig NEM. Arra nem is vállalkozhatunk, hogy ellenőrizzük, tényleg jól dolgozik-e a masina, mert senki nem hajlandó a számolást utánacsinálni. További problémája ennek a módszernek, hogy az egyenlőtlenségeket tartalmazó tételekkel csak nagyon nehezen, vagy egyáltalán nem boldogul, és a megoldás bonyolultsága igencsak függ attól a ténytől, hogy milyen módon és hány változót használva írjuk fel polinomjainkat. Jobb lenne, ha egyszerű halandó számára is felfogható bizonyításokat produkálna a program. Szerencsére már ilyen is van, de mielőtt e program bemutatásához hozzákezdenék, idézzünk fel néhány többé-kevésbé elfeledett dolgot!

Esetek

Ha még visszaemlékszünk a középiskolai anyagra, a geometriai bizonyításoknál gyakran szükség volt több eset vizsgálatára. Például a középponti és a hozzá tartozó kerületi szög kapcsolatának bizonyításához legalább három esetet kellett megvizsgálni. Bonyolultabb bizonyítások esetén ez a szám jócskán megnövekedhet. Hogy a gépnek ne kelljen mindannyiszor majdnem ugyanazt megcsinálnia, érdemes ezeket az eseteket egységesen kezelni, tehát olyan mennyiségekkel dolgozni, amelyek ezekben az esetekben ugyanazokat az eredményeket adják.

Chou és kollégái több ilyen, ún. invariáns fogalmat vettek górcső alá. A legkönnyebben felfogható fogalom talán az előjeles terület. Itt egy háromszög területe a magasság és az oldalhossz értékeken kívül a körüljárási irányától is függ. Így az ABC háromszög területe megegyezik a BCA és CAB háromszögek területével, de ellentettje (mínusz egyszerese) a ACB háromszög területének. A négyszögek területét két háromszög területeként értelmezzük, így az ABCD területét az ABC és ACD háromszögek területének összege adja. Azzal direkt nem foglalkozunk, hogy ez a négyszög esetleg nem elfajuló-e, mert a P és Q, valamint A és B pontok által meghatározott egyenesek (természetesen különböző pontokról van szó) párhuzamosságát pontosan az határozza meg, hogy a PAQB (esetleg elfajult) négyszög területe nulla-e vagy sem. Sok más fontos kritérium is megfogalmazható e speciális területfogalom segítségével. Lássuk, hogyan lehet ezt felhasználni a gyakorlatban!

„Háromszögelés"

Közismert, hogy a háromszög súlyvonalai egy pontban metszik egymást, és e pont harmadolja a súlyvonalakat. Ennek bizonyításához legyen ABC a vizsgálandó háromszög, F és E az AC, illetve BC oldalak felezőpontja, valamint O az AE és BF súlyvonalak metszéspontja. Nekünk elég azt bizonyítani, hogy az AO és OE szakaszok hosszának az aránya kettő. E két szakasz aránya megegyezik az ABF és BEF háromszögek területének arányával (ez egy könnyen igazolható tény, és mivel igen gyakran felbukkanhat ezekben a bizonyításokban, be lett építve a programba). Tudjuk továbbá azt, hogy a BEF területe fele a BCF területének, az ABF területe fele az ABC területének, valamint BCF területe fele az ABC területének. (Ha az oldal kétszeresére nő, és a magasság marad, akkor a terület is duplázódik.) Ezeket behelyettesítve az ABF és BEF arányba, már kapjuk is a kettőt eredményül. Ha ezt nem szövegesen írom le, hanem képletekkel, akkor az egész egy sorban is elfér. A hagyományos bizonyítás ennél egy picit hosszabb. Chou módszerének előnye ez a világos, könnyen érthető bizonyítás. Az általuk írt tételbizonyító program gyakran a gyakorlott geométereknél is frappánsabb, elegánsabb bizonyításokat készít.

E területfogalmon kívül Chou használja még a vektorokat, ahol szerepeltet két, ún. belső és külső szorzást, amelyek az előjeles területhez hasonlóan ugyancsak elég jó tulajdonságokkal rendelkeznek. Az előbbivel a merőlegességet, az utóbbival a párhuzamosságot vizs-

gálhatjuk meg könnyedén. Valójában a vektorok mint komplex számok szerepelnek, de ezt az egyszerű felhasználónak nem fontos tudnia, ezt a program tökéletesen elrejti a szeme elől. Említést érdemel még a *teljes szög* fogalma, amely a szögmértéket nem mint félegyenesek által bezárt szöget tekinti, hanem egyenesekkel dolgozik, és az a kérdés, hogy milyen elforgatás kell ahhoz, hogy a szög egyik szára a másikkal párhuzamos lehessen.

Nem az a bizonyos fix pont!

Talán külön cikket is megérdemelne a negyedik módszer, amelyet fölhasználnak bizonyítóprogramjukban, mert nagyon izgatja a fantáziámat. A módszer fixpont elnevezést kapott, de valójában speciális adatbázisról van szó, amely a megadott geometriai elrendezésben keres olyan objektumokat, amelyek az elrendezés transzformálása esetén is megmaradnak. Ez azért különleges, mert így nemcsak egy keresett objektum megléte vagy hiánya vizsgálható, hanem a program kiszűrhet olyan dolgokat is, amelyekre mi nem is figyeltünk fel, így a program akár új tételeket is önállóan felfedezhet. (Ez azért nem semmi!)

Noha a feladatok szövegében egyenesek, szögfelezők, körök szerepelnek, valójában csak pontokkal foglalkozunk, mert ezek a pontok is elegendőek az előbb említett objektumok meghatározására. E pontok három típusúak lehetnek: szabadok, félszabadok és kötöttek. Szabad pont például a tér egy tetszőleges pontja, tehát minden olyan pont, amelyre nincs semmilyen megkötés. A félszabad pontoknál már van bizonyos feltétel, de ez még nem határozza meg a pont pontos helyét, ilyen például a kör egy pontja, az egyenes egy pontja vagy egy adott ponton áthaladó adott egyenessel párhuzamos egyenes egy pontja. A kötött pontok természetesen azok, amelyeknek a helye egyértelműen adott, így például két egyenes metszéspontja, egy kör középpontja, a magasság talppontja, és így tovább. Egy (reménybeli) tétel megadása a következőképpen megy. Fel kell sorolni a feltételeket, azaz hogy a feladatban szereplő objektumok (pontosabban az azokat meghatározó pontok) hogyan is állnak elő, valamint meg kell adni, hogy mit is akarunk ezekből bebizonyítani. A feltételeket a HYPOTHESES, a bizonyítandó összefüggést a SHOW szó vezeti be, míg maga a feladat megfogalmazása az EXAMPLE után kezdődhet. Az összefüggések megadására igen sok lehetőségünk van (több mint negyven), ezért csak néhányat mutatok be közülük: a POINT A B C jelöli azt, hogy az A, B és C szabad pontok; CIRCUMCENTER O A B C pedig azt, hogy az O pont az A, B és C pontok által meghatározott kör középpontja. ON-CIRCLE D O A segítségével a D pont az O középpontú A-n keresztülmenő kör egyik pontja lesz. Ez utóbbit írhattuk volna ON-CIRCLE D (CIRC O A) formában is, mert van ilyen bőbeszédes megfogalmazás is, amely olvashatóbbá teszi a feladat szövegét, bár aki nap mint nap ezt a programot használja, biztosan a rövidebb alak mellett dönt.

A tétel állítása is majdnem húsz különféle alakot tartalmazó algebrai kifejezés lehet, így például az adott pontok egy egyenesre esnek, az adott pont az adott szakasz közepén található, az adott két egyenes párhuzamos. Van viszont egy olyan is, hogy NO, azaz a program nem bizonyít. Első látásra ez értelmetlennek tűnik, de azért van itt, mert a program csak grafikus felületen fut — Windows, illetve Openwindows alatt —, és nemcsak bizonyít, hanem rajzol is (pontosabban ebben az esetben csak rajzol). Miután megadtuk a szabad és a félkötött pontok helyzetét, a program felrajzolja a feladathoz tartozó ábrát. Mivel esztétikai rész nem lett beépítve a programba, így minden felhelyezett pontnál (PC-s verzió esetén) külön meg kell adnunk, hogy a pont betűjele hová is kerüljön.

Ha egy ábra nagyon kusza lett, nem kell az elejétől kezdeni a rajzolást, hanem az ábra egyes pontjait elmozgatva, odébbhúzva a program felrajzolja az új helyzethez illeszkedő ábrát. (Sajnos a PC-s verziónál a pontok címkéit kézzel kell a helyükre vinni az ilyen-olyan transzformációk után.)

Átlófelezés

Lássunk egy példát, amely azt állítja, hogy a paralelogramma átlói felezik egymást:

```
EXAMPLE Parallelogram

HYPOTHESES: POINT A B C;

      INTERSECTION_PP D C A B A B C;

      INTERSECTION_LL O A C B D;

SHOW: MIDPOINT O A C;
```

Ez magyarul a következőket jelenti: felveszünk három pontot, név szerint A, B és C-t. A D-t úgy kapjuk meg, hogy vesszük a C-n átmenő AB egyenessel párhuzamos, és az A-n átmenő BC egyenessel párhuzamos egyenesek metszéspontját. Az O-t a két átló (az AC és BD egyenesek) metszéspontja adja. A kérdés az, hogy az O pont felezi-e az AC szakaszt. Miután ezt átadtuk a programnak, pillanatok alatt megkapjuk a választ: „Ez igaz, feltéve ha A, B és C nem esik egy egyenesre, és ha AC nem párhuzamos BD-vel."

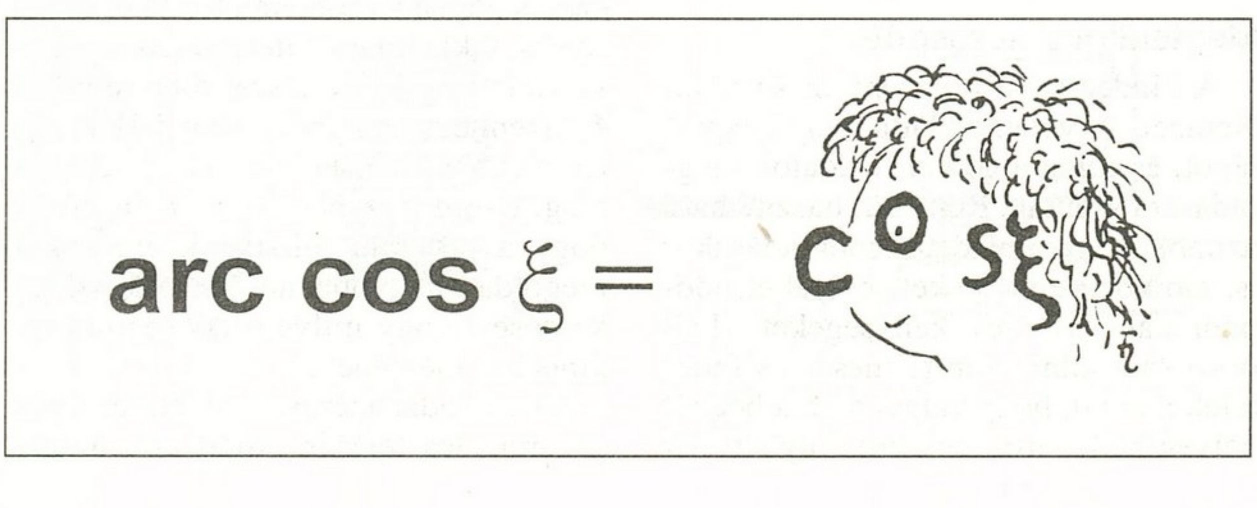
A kényelmesebbek nemcsak egy így megírt feladatot rajzolhatnak fel, hanem a feladatot is megfogalmazhatják az egér segítségével. (Bár könnyebb és kényelmesebb, de biztosan lassabb így megadni egy feladatot, mint begépelni.) Itt menürendszerből választhatjuk ki, hogy milyen művelet segítségével kapjuk az újabb pontokat a kiválasztottakból.

Azt csak a „beavatottak" értékelhetik, hogy a bizonyítás szövegét a program LaTeX formátumban elkészíti, és az ábrát PostScript formában kimenthetjük. Csupán azért nem lehet a „házi feladatot" így beadni, mert a program angol nyelvű, és a megoldást még le kell fordítani magyarra.

A grafikus felület miatt kicsit komolyabb erőforrást igényel a program. Windows alatt mindenképpen szükség van 8 Mbájt memóriára, ami annak is az eredménye, hogy a program többek között tartalmaz egy LISP interpretert.

Ahol hasznos...

A program eredetileg automatikus geometriai tételbizonyításra készült. Tehát ha valaki csupán arra kíváncsi, hogy egy adott állítás tétel-e vagy sem, akkor azt ennek segítségével könnyen megtudhatja. Remekül használható viszont taneszközként is, mivel a hat bizonyítási módszer bármelyikével



megpróbálhatjuk állításainkat igazolni, így ugyanarra más és más stílusú bizonyításokat kaphatunk, s igazán érdekes és tanulságos ezeket a bizonyításokat összehasonlítani.

Azt sem szabad lebecsülni, hogy a programmal könnyedén készíthetünk esztétikus ábrákat mindenfajta kézügyesség nélkül is, és azokat beilleszthetjük dolgozatainkba vagy tankönyveinkbe. A tanítási célokat szolgálja az is, hogy az ábrák könnyedén megváltoztathatók, és így ugyanazt a feladatot több szempontból is megvizsgálhatjuk. Ezt segíti elő az is, hogy a program képes animációra, azaz egy adott pontot megadott útvonalon mozgatva mindig újrarajzolja az ábrát. Ismerve a középiskolai oktatást, nem remélem, hogy matematikaórákon ez a program ugyanakkora szerephez jut, mint a derékszögű vonalzó, de az érdeklődő diákokat esetleg egy szakkör keretében meg lehetne vele ismertetni. Tanácsos a tanároknak is ez az eszköz (mint egy utolsó mentsvár), mert könnyen elveszíthetjük tekintélyünket, ha nem vagyunk képesek rövid idő alatt egy diákunk által valahonnan előbányászott geometriai feladat bizonyítását megalkotni.

Igazságtalan lennék, és megtagadnám magam, ha nem említeném meg, hogy szövegfájlból geometriai ábrákat készíthetünk a MetaFont és MetaPost segítségével is, és talán ezen programok eredményeit könnyebben illeszthetjük be dokumentumainkba. Továbbá az Alaplapban már szerepelt egy olyan oktatóprogram leírása, amelyben a geometriai objektumokat hasonló módon adhattuk meg, és ott is volt lehetőség az animációra (Alaplap, 1992. február). Az viszont egyértelmű, hogy ennek a programnak a bizonyítás terén nincsen párja. (Azért ez a program sem mindenható, mert vannak olyan állítások, amelyeket nem tudunk megfogalmazni az adott geometriai nyelven.)

Ha valakinek ez a program felkeltette az érdeklődését, akkor az em-city.cs.twsu.edu ftp-címen a pub/geometry/software alkönyvtárban megtalálhatja minden verzióját, több mint 80 oldalas dokumentációját, valamint az itt ismertetett formától kicsit eltérő módon rögzített 500 tesztállítását. Ugyanezen a gépen megtalálhatók a program szerzőinek mindazon publikációi is, amelyekben megmutatják, hogyan lehet használni ezeket a módszereket a térgeometriában, a nem-euklideszi geometriákban, illetve a differenciálgeometriában.

Aszalós László

A kivétel (nem) erősíti a szabályt!

Két helyen lenni

Annak bizonyítására, hogy valami nem tétel (vagyis mint állítás nem állja meg a helyét), választhatjuk módszerként egy ellenpélda megmutatását. Az ilyen ellenpéldák megtalálása viszont nem egyszerű feladat. Sok száz, vagy még gyakrabban több ezer esetet kell végignézni, hogy megfelelnek-e az adott kritériumnak, vagy sem.

Sosem értettem, miért lenne valami szabály, ha bizonyos esetekben nem teljesül. Szerencsére a matematikában ilyen nincs. Ott valami csak akkor szabály, ha mindig teljesül, azaz nincs semmilyen kivétel.

A sokat emlegetett négyszínproblémának (azaz kiszínezhető-e minden térkép négy színnel úgy, hogy a szomszédos országok eltérő színűek legyenek) a bizonyítása úgy történt, hogy a többezernyi, elvileg különböző lehetőséget egy számítógép végigpróbálgatta, és megállapította, hogy minden esetben kiszínezhető a felület négy színnel. Ekkor hördült fel a matematikai közvélemény, hogy miféle bizonyítás ez, sőt egyáltalán nevezhető-e bizonyításnak. (És azt tartalmazza-e a program, amit a szerzője meg akart írni? Vagy a program valóban úgy fut-e le, ahogy mi elképzeljük?)

Mi nem bonyolódunk bele ilyen filozófiai problémába, hanem megismerkedünk egy olyan programmal, amely ugyancsak végigrágja magát a lehetőségeken, de nem bizonyítást, hanem ellenpéldát keresve. Ha találunk egy ilyent, akkor a sejtésünk megdőlt, viszont az ellenpéldát vizsgálgatva rájöhetünk, hogy hol volt hibás az elképzelésünk. Ha a program nem akad ellenpéldára, akkor valószínűsíthetjük, hogy elképzelésünk helyes, de ezt azért nem árt bebizonyítani a megszokott eszközökkel is.

Megtalálni a keresettet

A Finder nevű program az Ausztrál Nemzeti Egyetemen látta meg a napvilágot, és főleg algebrai feladatok megoldására szánták. Remekül használható azonban olyan feladatok megoldásakor is, amikor személyeket, helyeket, időpontokat és tevékenységeket kell összekapcsolni azon természetes feltételek mellett, hogy valaki nem lehet két helyen egyszerre, és nem folyhat egy helyen egyszerre két egymást kizáró tevékenység.

Talán a legismertebb ilyen példa az iskolai órarend elkészítése. Erre már születtek programok nagy számban, és az Új Alaplap is bemutatott közülük néhányat (Új Alaplap 1994. február). Noha még egyiknek a forráskódjába sem néz(het)tem bele, meg vagyok róla győződve, hogy a megszokott backtrack módszerrel működnek: „helyezzünk el egy órát, és teszteljük az összes feltételt" (esetleg a hátizsákproblémához hasonlóan az órák kapnak egy súlyozást, hogy melyikkel kezdjük el a próbálkozást). Ezzel a módszerrel az a probléma, hogy az összes lehetőségek száma ugrásszerűen (precízebben szólva exponenciálisan, azaz hatványozottan) függ a probléma méretétől.

Az órarendproblémához kísértetiesen hasonlít a nyolc királynő problémája, amikor is egy sakktáblára úgy kell felrakni a nyolc királynőt (vezért), hogy azok ne üthessék egymást. Mivel tudjuk, hogy a királynők azonos sorban vagy azonos oszlopban nem szerepelhetnek, már csak a 8! (nyolc faktoriális) — azaz nagyjából negyvenezer — felállásból kell kiválasztani azt a közel kilencvenet, amelyben átlósan sem ütik egymást a bábok. Noha most nincs mihez viszonyítani, íme néhány futási eredmény: ez a program az összes megoldást fél másodperc alatt megtalálta egy Sparc munkaállomáson. Ugyanezen a gépen a probléma tizenkét mező széles sakktáblára kiterjesztett variánsát (ami közel tízezerszer több felállást és tizennégyezer megoldást jelent) kilencvenhét másodperc alatt oldotta meg. (Nem hagyható figyelmen kívül, hogy a program pillanatok alatt talál megoldást nagyobb táblák esetén is, így kérdéses, hogy milyen időtartamot tekintsünk mércének.)

Ez a feladat azért sem alkalmas ilyen program tesztelésére, mert erre a spe-

ciális problémára vannak olyan algoritmusok, amelyek rövid idő alatt könnyedén megbirkóznak akár milliónyi királynővel is.

Egy kis ügyességgel ezt a programot hagyományos logikai feladatok megoldására is lehet használni, amelyben mindenki mond egy igaz és egy hamis állítást, és ezekből kell megállapítani, hogy melyik igaz és melyik nem. Rejtvénykészítők is bizalommal forgathatják ezt a programot, mert nincs annál nagyobb égés számukra, mint amikor kiderül, hogy rejtvényüknek nem csak egyetlen megoldása van. Ez a program viszonylag gyorsan eldönti, van-e esetleg másik megoldás is.

„Halmozzuk az élvezeteket"

Ha sikerült felcsigázni az érdeklődést, akkor „jöjjön a feketeleves". Egy kicsit fel kell eleveníteni algebrai ismereteinket. Véges halmazokkal dolgozunk, és egyszerre több halmazunk is lehet. E halmazok megadásakor méretüket (a számosságukat) kell meghatároznunk. Ha nekünk úgy kényelmesebb, akkor minden elemet külön-külön is elnevezhetünk. E halmazokon különféle függvényeket (pontosabban műveleteket) adhatunk meg. Meghatározhatjuk a függvény típusát, azaz hogy honnan és hová hat, mindenütt értelmezett legyen-e, és további speciális tulajdonságokat is megadhatunk.

Ezek után következnek a modellel szembeni elvárásaink, azaz hogyan is viselkedjenek a függvényeink. Ekkor szabhatjuk meg azt is, hogyan dolgozzon a program. Több tucat paramétert állíthatunk be, amelyeket most teljesen felesleges felsorolni. Aki megszerzi a programot, az úgyis megtalálja benne a dokumentációt, amelyben minden részletesen le van írva.

Lássunk egy egyszerű példát, amely bemutatja az alapokat (habár nincs semmi gyakorlati jelentősége): legyen négy elemünk, az egyszerűség kedvéért nevezzük el az ábécé első négy betűjével, és ez a sorrend legyen a rendezés alapja is, valamint legyen ezen a halmazon egy szorzással jelölt művelet, amely asszociatív (tehát szabadon zárójelezhető), és mind elölről, mind hátulról szorozva megtartja a rendezési relációt. Az ennek megfelelő Finder input:

```
1 sort element { enum: a, b, c, d. }
2 function *: element,element ->
  element {}
3 clause {
4   x < y,  x * z > y * z -> false
5   x < y,  z * x > z * y -> false
6   x * y * z = x * (y * z).
7 }
8 end
```

A sorok elején álló számok nem részei az inputnak, itt csak azért vannak, mert így könnyebben tudom elmagyarázni az egészet. Az 1. sor határozza meg az „element" típust, és elnevezi a négy elemét. A 2. sor a szorzás függvényt vezeti be, megadja, hogy ez kétargumentumú, az argumentumai „element" típusúak, és a függvény értéke is ilyen típusú lesz. A 3. sorban kezdődik a tulajdonságok felsorolása, és a hetedik sor zárja le azt. A nyolcadik sor jelzi, hogy vége van a meghatározásnak. A 6. sor az asszociatív tulajdonságot írja le, itt fel lett használva a zárójel előtt, hogy ha másként nem mondjuk, a függvényeink bal asszociatívak. A 4. és az 5. sor pedig állítja: lehetetlen, hogy bal vagy jobb oldali szorzással megforduljon a reláció.

Az e feltételeknek megfelelő 316 modellből az egyik:

	a	b	c	d
a	a	a	a	a
b	a	a	b	b
c	a	b	c	d
d	a	b	c	d

Itt tizenhat helyre lehet négy-négy betűt beírni, és erre összesen közel négymilliárd lehetőségünk lenne.

Lássuk most a királynős feladatot, és figyeljük meg, hogy mennyire egyszerű ez a megfogalmazás:

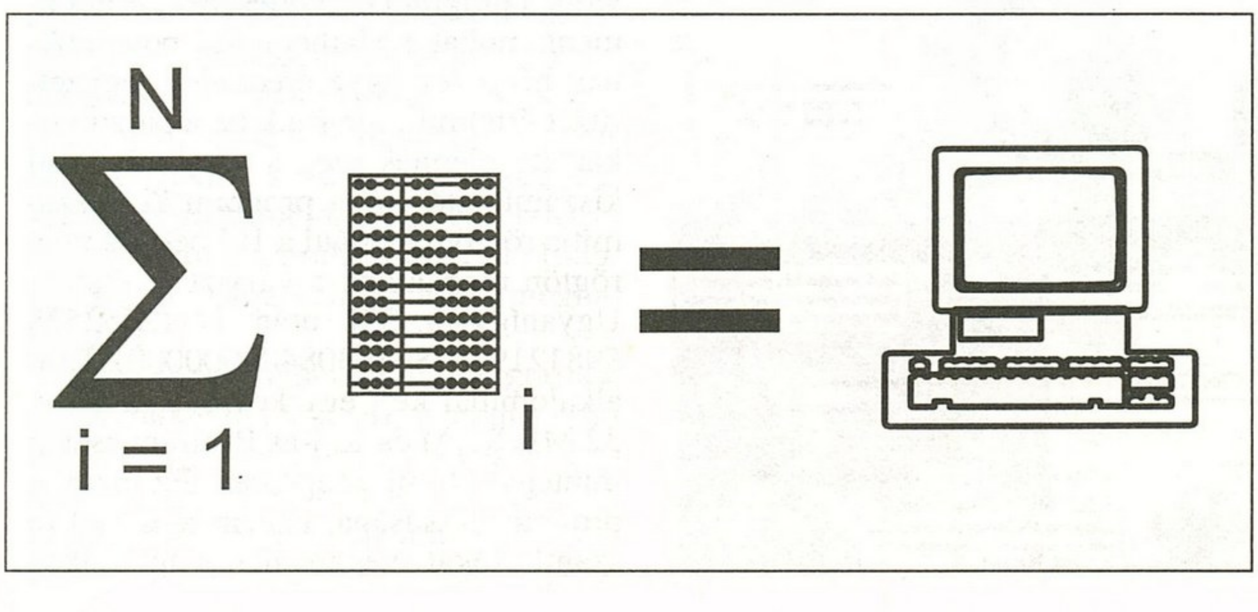
```
1 sort {
2   sor       cardinality = 10
3   oszlop   cardinality = 10
4 }
5 function queen: sor -> oszlop
6   bijective
7 clause {
8     E!(r+x),
9     queen(r)+x = queen(r+x) -> x=0
10    E!(r+x),
11    queen(r) = queen(r+x)+x -> x=0
12  }
13  setting {
14     primary
15     verbosity models: none
16  }
17 end
```

Mit is csináltunk? Hogy könnyebb dolgunk legyen, nem sakktáblával dolgozunk, hanem egy olyan függvénnyel, amely megadja, hogy az adott sorban álló királynő melyik oszlopban található. A bijektív tulajdonság megkövetelése kizárja, hogy egy sorban vagy egy oszlopban egyszerre két figura is legyen. Ezek után már csak az átlókra kell majd figyelni. Az input második és harmadik sora határozza meg a méreteket. Az E! egy speciális függvény, azt adja meg, hogy létezik-e az argumentumában szereplő elem. Itt most egy összeg van, ami tulajdonképpen az r-et x-edikként követő elemet jelenti, illetve esetünkben azt, hogy a keresett elem a táblán van-e még. A kilencedik és a tizenegyedik sor pedig azt zárja ki, hogy két bábú átlósan helyezkedhessen el. A beállítások nem különösen érdekesek, a „primary" azt jelenti, hogy a kétszintű keresésből a második szintet ne használja. Ennek segítségével az összes megoldás 66 másodperc alatt is meglesz.

A cikk méretének korlátai miatt (és hogy az absztrakt algebrában járatlanokat nehogy elriasszuk) ilyen egyszerű feladatot választottunk. Ez azonban nem jelenti azt, hogy komolyabb feladatok megoldására ne lehetne ezt a programot jól használni. Van jó néhány olyan, mai napig meg nem oldott, komoly matematikai probléma, amelynél esetleg sikert érhetünk el vele.

A lemezmellékletre rátettünk néhány hosszabb, szórakoztató feladatot, hogy pontosabb elképzeléseink lehessenek erről a programról. Már csak az érdekesség kedvéért is, hogy lemérhessük, hányszor gyorsabb a gép nálunk!

Aszalós László



Ma és Ma

Két „mindentudó matekos"

A számítógépet, mint tudjuk, matematikusok találták fel, sőt, kezdetben elsősorban matematikai feladatok megoldására használták: differenciálegyenletek gyors numerikus megoldására, ami a háborúban igen fontos dolog volt. Csak később lett a domináns felhasználási terület az adatbáziskezelés, és sokkal később vált uralkodó alkalmazássá a szövegszerkesztés. Ez azonban nem jelenti azt, hogy ne lettek volna olyan programok, amelyek matematikai feladatok megoldására szolgáltak.

Matematikai feladaton két, némileg különböző dolgot kell érteni. Az egyik a numerikus problémák megoldása, ekkor a számítógép gigantikus, gyors és különösen intelligens szuperszámológépként működik. A kezdetek egyik uralkodó programnyelve, a Fortran éppen ilyen feladatok megoldására volt a legalkalmasabb. A másik, amit ma a zsargon szimbolikus matematikának hív, és amit a gimnazista vagy a tudós egyszerűen matematikának nevez: ekkor a számítógép tulajdonképpen nem számol, hanem formalizált matematikai tudást alkalmaz a számára érthetővé tett jelsorozatokon.

Túl az egyenleten

Már a PC hőskorában is létezett az Eureka és a Tksolver, mind a kettő képes volt például egyenleteket és egyenletrendszereket megoldani. A komoly szimbolikus matematikához azonban komoly programozói munka kellett (ez akkoriban még megvolt) és „erős" számítógép, ez viszont akkoriban még csak „relatíve" volt meg. Beletelt néhány évbe, míg a PC-re is került (átírva vagy eleve arra szánva) igazi matematikai program. Ezek — talán legfőbb — közös jellemzője, hogy nevük szinte mindig az Ma betűsorozattal kezdődik: Mathcad, Mathematica, Maple, Matlab, Macsyma.

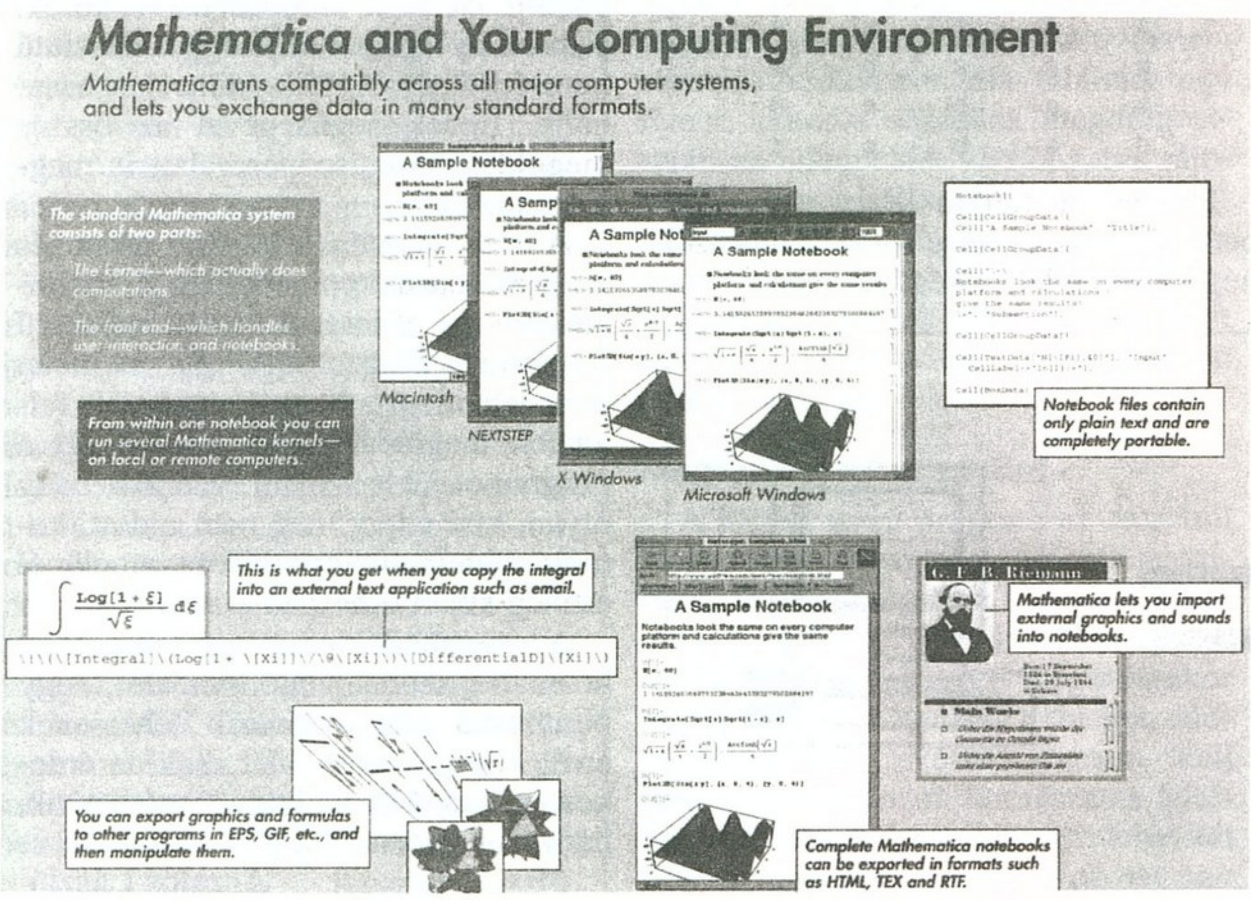
A Mathcad elsősorban mérnöki eszköz, nagyon tud számolni, rengeteg kézikönyvbeli képletet és együtthatót ismer, és egy kicsit tudja a szimbolikus matematikát is — az utolsó, általam ismert változat éppen a Maple egy lefaragott részét tartalmazta, világképe — ha lehet ilyet mondani egy szoftverről — a sárga munkafüzet, amely él és reagál.

A Matlab mindent mátrixként kezel — ezt lehet fogyatékosságnak tekinteni, de a program ezzel a megközelítéssel is igen intelligens. Nem a szimbolikus matematikában erős, bár az általam látott változat 3 éves, azóta ez is fejlődhetett.

A három nagyágyú, a Mathematica, a Maple és a PC-n újnak számító Macsyma közül csak az első kettőt tudtam megnézni, ezek viszonylag friss, egy évnél nem régebbi verziók. Már itt le kell szögezni, hogy komoly programokról van szó! Megtanulásuk — ha a triviális használaton túl akarunk lépni — igen időigényes. Ezért ez a beszámoló szükségképpen felületes lesz. Amiről azt írom, hogy megvan ezekben a programokban, az megvan, de amiről nem írok, az is ott lehet, csak esetleg nem találtam meg. Mind a két szoftvert használják kutató matematikusok: nem pótolja az agymunkát, de bizonyos részeit megkönnyíti, meggyorsítja. Persze használhatók középiskolai, egyetemi feladatok megoldására is, de aki eljut oda, hogy a feladatot át tudja fogalmazni a Maple vagy a Mathematica nyelvére, az rendszerint már ezek nélkül is meg tudja oldani.

Mathematica

A Mathematica eredetileg egy Wolfram nevű (nem kémikus és groteszk módon nem is matematikus) fizikus műve, mára nagyon komoly szoftverré nőtte ki magát. Installálás után 120 MB helyet foglal, szereti a nagy, erős gépet sok memóriával. Ha egy kicsit még nagyobb és még erősebb a gép, és még több a memória benne, az csak javára válik a program működésének. A dokumentumokat a Mathematica notebooknak hívja (ez ugye eredetileg jegyzetfüzet értelmű), ide írjuk be a parancsokat, itt jelennek meg a válaszok. Ami kiszámítható, azt a program ki is számítja rögtön. Például a 10! beírása után rögtön megkapjuk a választ (362880). Ugyanígy a 30! után is (2652528 598121910586363084800000000). Első alkalommal kell egy keveset várni — 32 MB RAM és 133-as Pentium esetén mintegy 10 másodpercet. Ez nem a program lassúsága, hanem a kernel (a számításokat végző mag) moduláris



$$\int \frac{Cos[x]}{Sin[x]^6 Sin[2x]^{1+½}}\,dx = -\frac{1}{390} Csc[x]\,(128 + 32Csc[x]^2 + 20Csc[x]^4 + 15Csc[x]^6)\,\sqrt{Sin[2x]}$$

1. ábra

szerkezete miatt van. A szoftver maga menedzseli a különböző részek behívását és kirakását, a megjelenítő, interfészmodul elválik a kerneltől. Így például lehet, hogy egy másik gépen fut a kernel, mint ahol a parancsokat beadjuk, és az eredményt látjuk.

A 30 faktoriális már elég nagy szám, több mint 30 jegyű, de a programnak ez nem okoz gondot. Az sem, hogy prímtényezőkre bontsa, a betöltött modul mellett ez érzékelhetetlenül kevés idő. Persze egyenletet is könnyű megoldani, és a megoldás, amikor csak lehet, nem numerikus módszerekkel, hanem „igazi" matematikával történik. A 2sin(x)cos(x)=3 megoldása sem tart sokáig, az eredményből pedig látszik, hogy a szoftver a komplex számoktól nem riad vissza (2. ábra).

$$Solve[2Sin[x]Cos[x] == 3, x]$$

$$\{\{x \to -ArcCos\,[\tfrac{1}{2}(-2-I\sqrt{2})]\}, \{x \to -ArcCos\,[\tfrac{1}{2}(2-I\sqrt{2})]\},$$
$$\{x \to -ArcCos\,[\tfrac{1}{2}(-2+I\sqrt{2})]\}, \{x \to -ArcCos\,[\tfrac{1}{2}(2+I\sqrt{2})]\}\}$$

2. ábra

Azonban az is látszik, hogy az output — bár logikus és érthető — kissé furcsán néz ki. Tulajdonképpen az a legnagyobb bajom a Mathematicával, hogy szintaxisa számomra idegesítő. A megszokás azt íratja az emberrel, hogy sin(x), és még véletlenül sem Sin[x]. A program sejti, hogy mire gondolok, figyelmeztet is, de meg nem eszi a hagyományosabb jelölést. (Bár lehet, hogy van valahol a sok száz között egy olyan opció, amivel ez is szabályozható.) A Mathematica megbirkózik mindennel, ami matematika. Elképesztő tudás van benne felhalmozva speciális függvényektől kezdve a geometriáig. Az egyik mintapéldán egy elborzasztó határozatlan integrált lehet látni (1. ábra). Ezt próbálja valaki kézzel meghatározni!

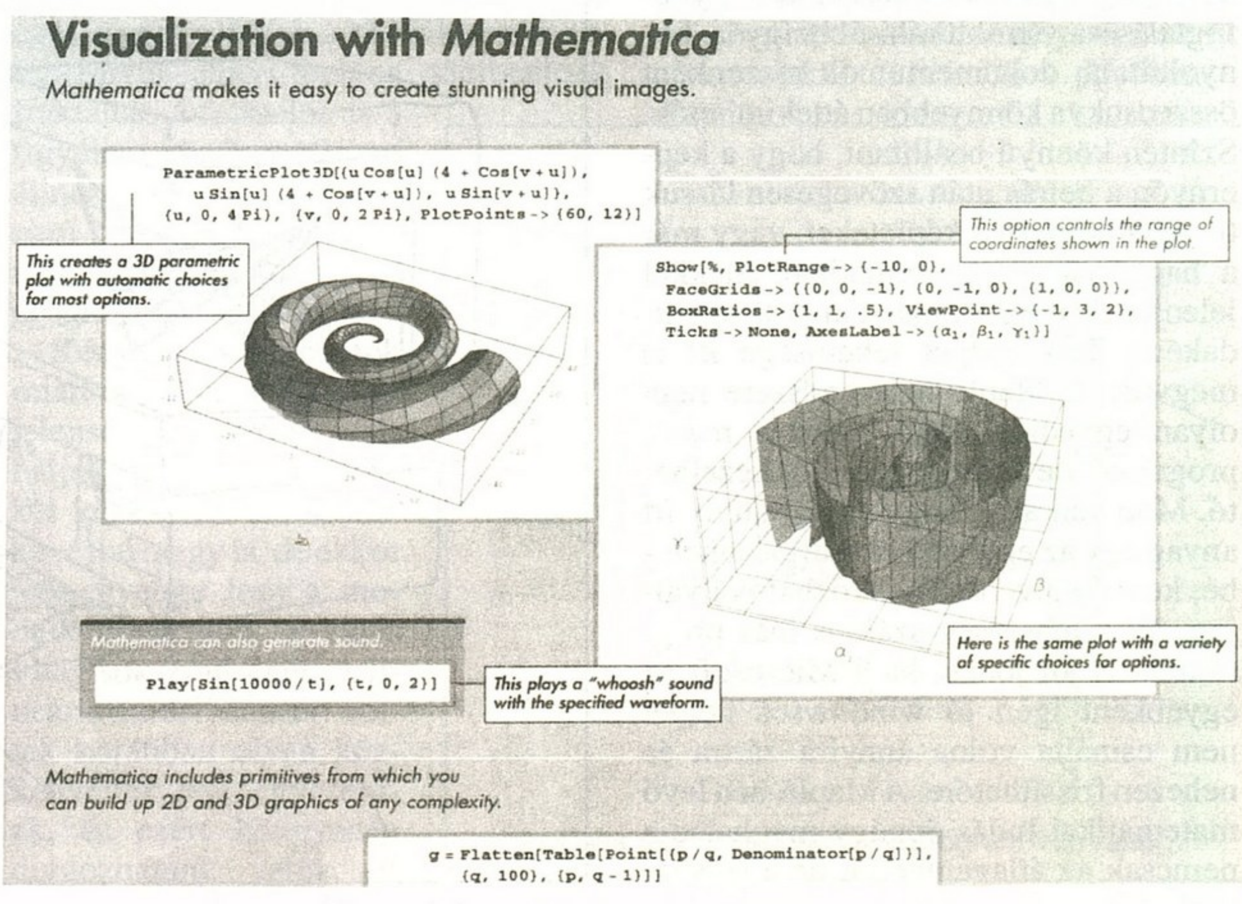
A szoftver hihetetlenül erős grafikai eszközökkel rendelkezik, a lapunk címlapján és az összeállítást bevezető írás mellett közölt illusztrációk is mutatják, hogy háromdimenziós ábrákat — beépített forgatási lehetőségekkel — játszi könnyedséggel készíthetünk vele. Ezek az ábrák sokszor nemcsak szépségük miatt érdekesek, hanem valóban sokatmondóak a hozzáértők számára. Az már külön többlet, hogy az ábrák animálhatók, így egy paraméteres függvény képe a paraméter változásával remekül nyomon követhető.

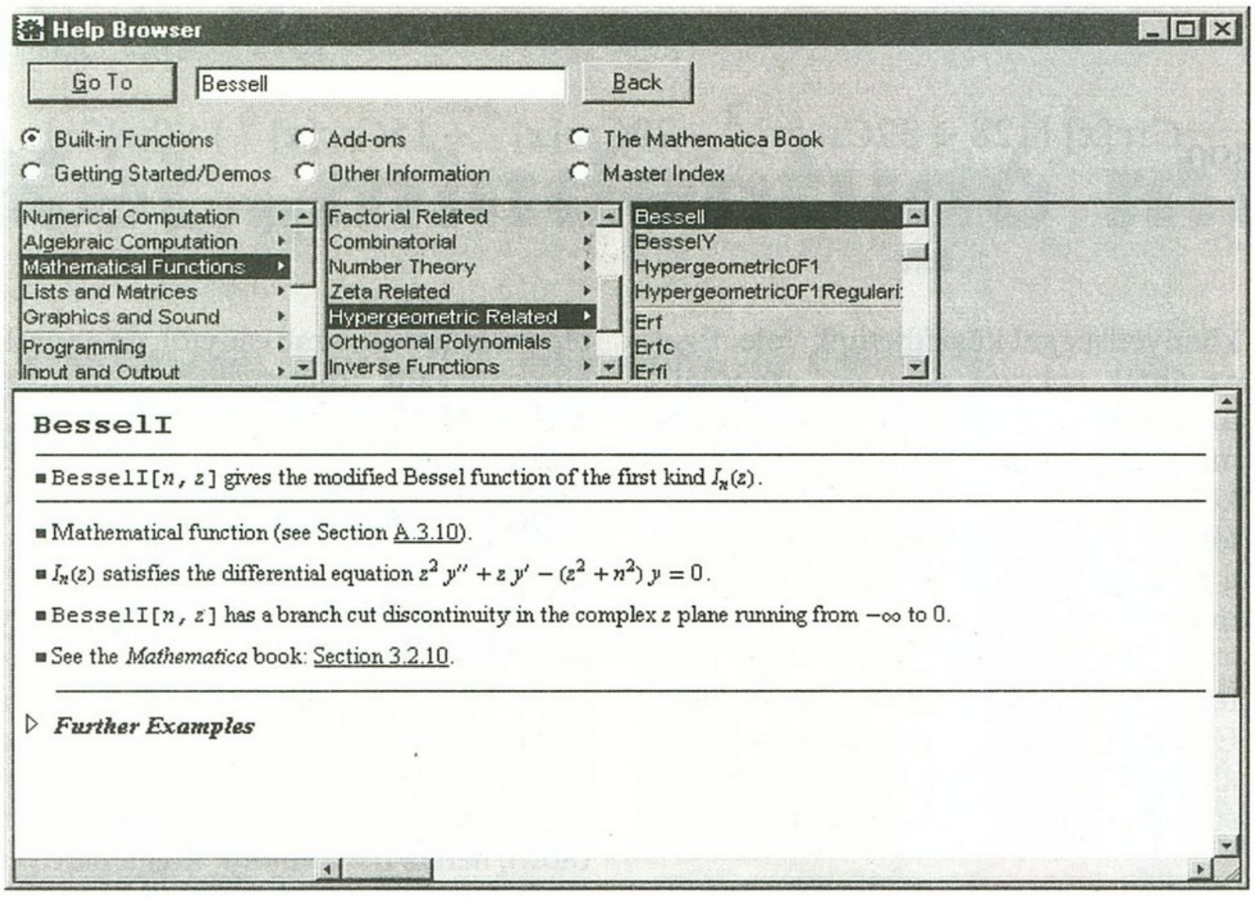
A Mathematica nagyon nagy súlyt fektet az eredmények megjelenítésére. Egy cikk nyomdakész változata is előállítható — méghozzá az előre definiált stílusok segítségével — pillanatok alatt. A nyomdakész szó szerint értendő, mert a program TeX outputot is képes gyártani, márpedig a szakfolyóiratok szeretik ebben kapni az anyagokat. A mai, Internettel „fertőzött" világ is megjelenik, az output lehet HTML formátumú, és akkor mehet egyenesen a Webre.

A help ragyogó és jól használható, moduláris felépítése és megjelenése eltér a windowsos szabványtól, de ez csak előnyére válik. Regényeket lehetne még írni arról, hogy milyen matematikai részterületek legfrissebb eredményei vannak integrálva a szoftverbe, hogy saját programnyelve van, hogy az Interneten és másutt tömegével lehet hozzá kész jegyzetfüzeteket találni, hogy még magyarul is van hozzá könyv. (Szili László és Tóth János Matematika és Mathematica c. művének ismertetése az Új Alaplap 1997. februári számában olvasható.) A Mathematica cseppet sem olcsó, nehéz megtanulni, drága hardver kell az elviselhető sebességű működéséhez, de csodálatos eszköz.

Waterloo és mégsem Waterloo

A Maple is nagyon híres program, fejlesztői a kanadai Waterloo Egyetemen dolgoznak, a szoftver neve is a kanadaiak szimbóluma, a juharlevél. A Maple sokkal kevesebbet kíván a géptől, mint a Mathematica, de azért elfoglal vagy 30 MB helyet, és szintén előnyben részesíti a sok memóriát. Felülete spártaibb, de még így is barátságosabb, mint a másik programé. A faktoriálisos kísérletet itt is elvégezhetjük, az eredmény nem meglepő módon ugyanaz, a megjelenése azonban kedvezőbb a szemnek, szinte olyan, mint a matematikakönyvekben. Itt egyéb-





ként nincs a Mathematica esetében említett várakozás, a Maple belső szervezése valószínűleg eltérő. Az előző integrálos feladattal azonban a Maple nem tudott megbirkózni, nem talált zárt alakot.

Könnyebb megadni a feladatokat, a szinuszfüggvény tényleg sin, az argumentumot kerek zárójellel lehet írni, csak a sorvégi ; az, amit nem célszerű lefelejteni. A Maple készítői sokkal kevesebbet foglalkoztak a felhasználói felület szépségével, viszont így a szoftver átláthatóbb és könnyebben kezelhető lett. Azért persze itt is van lehetőség „csicsázni", mintha szövegszerkesztőben lennénk. Hasonlóan a másikhoz (bár ott ezt nem említettük) mód van az anyag különböző részeinek egybefoglalására, strukturálására, így a bonyolultabb dokumentumok részenként összecsukva könnyebben áttekinthetők. Szintén könnyű beállítani, hogy a képernyőn a beírás után szövegesen lássuk (és javíthassuk) a képleteket, vagy már a hagyományos matematikai jelekkel jelenjenek meg. A már említett nyomdakész, TeX output lehetősége itt is megvan. A Maple helprendszere nem olyan egyedien átütő, mint a másik programé, de azért nagyon jól kezelhető. Mód van arra, hogy egy általunk írt anyag egy az egyben bekerüljön a helpbe, kereshetővé és hivatkozhatóvá váljon. Ez a lehetőség számos más programnál is jól jönne, ha a Microsoft az egyébként igen jó windowsos helpet nem csinálta volna annyira zártra és nehezen frissíthetőre. A Maple-ben levő matematikai tudás éppúgy meghaladja nemcsak az átlagemberét, de a specialistáét is, mint a Mathematica esetében. Találunk csoportelméletet, Lie-féle szimmetriákat (bármi is legyen az), különösen bőséges és szép geometriai és gráfelméleti anyag van benne, egy példa az outputok közül a mellékelt képen látható.

A Maple sem veti meg a grafikus ábrázolást, két- és háromdimenziós függvényrajzolási lehetőségek szép számmal vannak a programban. Talán az output nem annyira könnyen és elegánsan módosítható, mint a másik programnál, de az eredmény, láthatjuk, így sem hagy sok kívánnivalót maga után. Itt is lehet animált ábrákat készíteni.

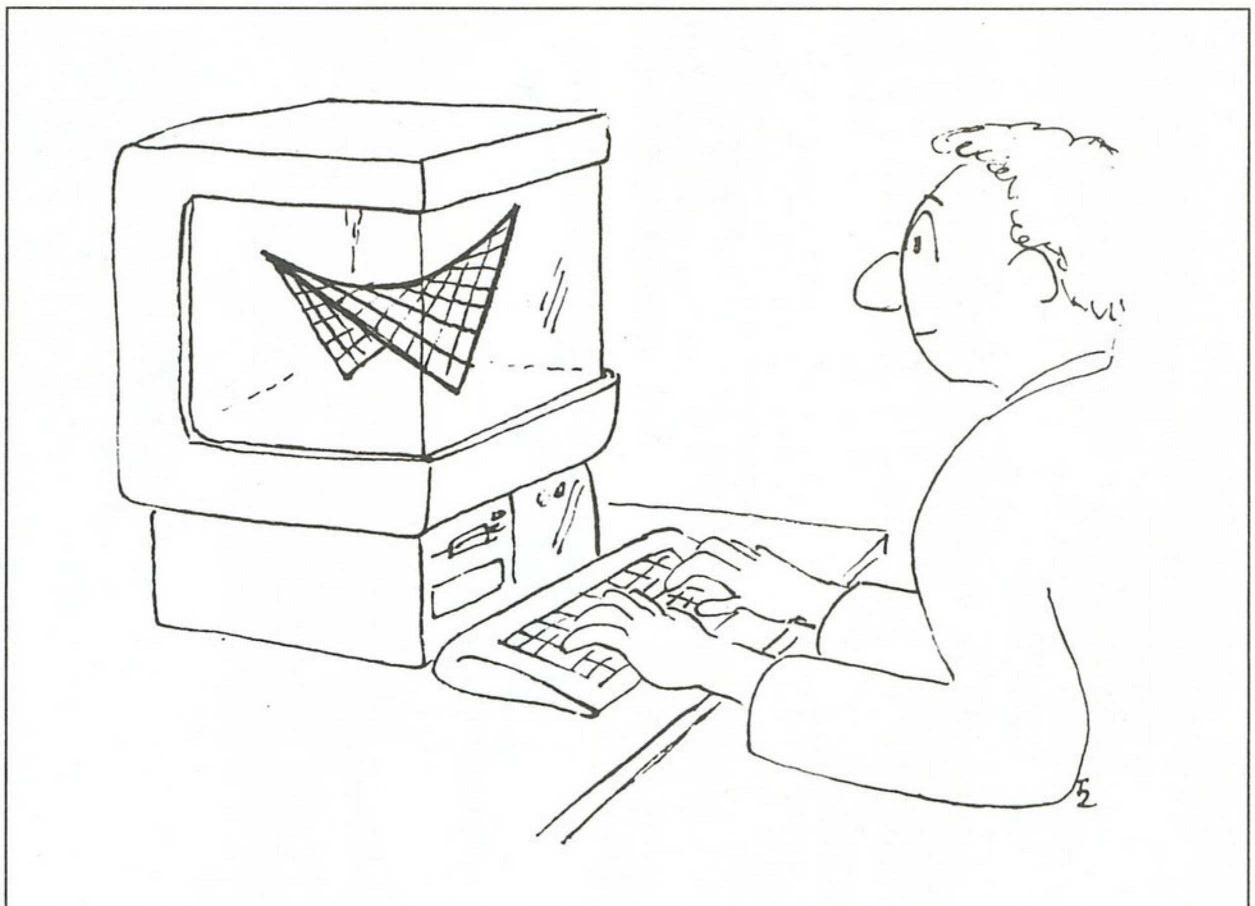
Mind a két programnak saját (természetesen teljesen eltérő) programnyelve van, ennek segítségével a már meglevő anyag bővíthető, speciális feladatokra, elméletekre könnyedén lehet megoldásokat programozni. Az újabb bővítések könyvtárba kerülnek, és a program és a felhasználó szempontjából egyaránt a szoftver integráns részévé válnak.

Magyar vonatkozások

Sok helyen tanítják egyik vagy másik programcsomagot, hazánkban is, többek között az ELTE-n vagy Gödöllőn. Tanulása nagyon hasznos, egyszerre segít a matematika, a számítástudomány és a programozás megértésében. Aki megtanulja valamelyiket, az nem nagyon fog a másikra átszokni, mert a triviálison kicsit is túlmutató használat nagy energiabefektetést igényel. Már egy szövegszerkesztőnél is kétszer meggondolják az emberek, hogy áttérjenek-e egy másikra, itt ez fokozottabban érvényes. Ezért a diákok, egyetemisták mind a két szoftvert igen kedvezményes áron szerezhetik meg (lásd még: beetetés). Ebben az esetben azonban ezt a gesztust csak dicsérni lehet.

Ha választanom kellene, bajban lennék. Valószínűleg a Mathematica a nagyobb és teljesebb, hozzám mégis a Maple állt közelebb. De azért nem hagyhatok ki egy zárógondolatot. Van a gépemen még egy matematikai program. A neve Derive, DOS alatt fut, összesen 450 K-nyi helyet foglal, jó neki a 286-os gép is. Nem tud annyit, mint a fentiek, de nem tud *sokkal* kevesebbet.

Horlai János



Determinisztikus? Sztochasztikus?

Ritmus és káosz

A természetben nemcsak ritmusos jelenségek vannak, hanem többé-kevésbé kaotikusak is. A matematika kialakulása valaha a ritmusok, a „szabályosságok" feltárásával indult el, és művelői csak jóval később kezdtek el komolyabban foglalkozni az ettől alaposan eltérő jelenségek megismerésével...

A környezetünkben igen sok dolog ciklusosan változik. Gondoljunk csak a napok, éjszakák, vagy akár az évszakok váltakozására. De nemcsak az élettelen, hanem az élő természet is időről időre változik, az élő szervezeteknek is vannak ciklusai. Gyakran nehéz észrevenni, hogyan is alakulnak ki ezek az egymást ritmusosan követő események. Mondjuk a színházban az előadás után a kósza tenyércsapásokból pár másodperc alatt a ritmusos vastaps. Ennél jóval látványosabbak a dél-ázsiai szentjánosbogarak, amelyek természetesen nem tapsolnak, hanem fel-felvillannak, de nem úgy elvétve, egymástól függetlenül, hanem akár több ezer is egyszerre, egy ütemben, mintha egy láthatatlan karmester irányítaná őket.

Persze ne gondoljuk azt, hogy a ritmusos, periodikus jelenségek mindig azonnal felismerhetők. Ki gondolná, hogy a feldobott kő pályájának leírásában is van ritmus? Pedig a mozgás másodpercenként leolvasott, szomszédos függőleges koordinátái különbségeinek különbsége már konstans érték, amit szokás g-vel is jelölni, és e ritmus felfedezése vezetett el a differenciálszámításhoz.

A most bemutatásra kerülő dinamikus rendszerek vizsgálatához komolyabb matematika szükséges, és sok számolás, ezért nem is csoda, hogy csak napjainkban foglalkozhatunk velük érdemben.

Linearitás

Ennek a tudományágnak egyik megszokott jelzője a nemlineáris. Bár a tagadásnak általában van valami rosszalló hangulata, ez a matematikában egyáltalán nem érvényes. A matematikában a tagadás inkább a régi, megszokott fogalmakon lép túl, és újabb, érdekes területekre visz. Napjaink egyik felkapott területe például a nemstandard analízis, amely szakít az elmúlt századok epszilonisztikájával, és a newtoni alapokról kiindulva, a modellelméletet felhasználva könnyedén old meg olyan feladatokat, amelyekkel az ún. standard analízis csak nagyon nehezen birkózik meg.

A linearitás hivatalos definíciója az $f(x+y)=f(x)+f(y)$ és az $f(a*x)=a*f(x)$ feltételeket követeli meg. Mindaz, amire ez nem teljesül, nemlineáris. Ha szemléletes képben akarjuk érzékeltetni: lineáris az, ami közelről és távolról is ugyanúgy néz ki, tehát ha valamiről első látásra nem látszik, hogy mekkora. Például egy elefánt nem lehet akkora, mint egy egér (olyan lábakkal meg sem mozdulhatna), és egy egér sem lehet elefánt méretű (mert fel sem tudna állni). Itt szót ejthetünk a fraktálokról is, amelyeket több módon is szokás meghatározni. Az egyik szerint fraktál mindaz, amit tetszőlegesen közelről nézve sem látunk lineárisnak. Vannak ezek között egyszerűbbek, bonyolultabbak, de legismertebbek a felhők, hegyek, tengerpartok, a fa gyökerei és ágai, vagy mondjuk a tüdőnk szerkezete. Mivel ennyi helyen szerepelnek fraktálok, a fraktálok is a figyelem középpontjában állnak, de szerencsére nem csak képernyővédőnek vagy tesztelő programnak használják.

Kényelmi szempontok miatt a valós környezet jelenségeit gyakran próbáljuk lineáris modellekkel közelíteni. Ez többé-kevésbé megy is, de ekkor igen gyenge lesz a modell, elveszítjük a valós világ sokszínűségét. A lineáris modellek azért voltak korábban olyan közkedveltek, mert egyszerűek, és ezért könnyedén dolgozhatunk velük. A számítógépek megjelenésével viszont a számolási kapacitások elképesztő mértékben megnőttek. Aminek kiszámítására korábban egész emberöltő kellett, azt ma a gépek napok vagy órák alatt elvégzik. Ezzel lehetővé vált, hogy a valós világot jobban megközelítő modellekkel dolgozhassunk.

Dinamikus rendszerek

Másik fogalmunk a dinamikus rendszer. Itt a rendszer állapota pontosan leírható néhány értékkel, és vannak olyan dinamikus szabályaink, amelyek ezen állapothoz megadják az elkövetkező időpontok állapotait. Ha ezeket egyértelműen lehet megadni, akkor determinisztikus rendszerről beszélünk, ha nem, akkor sztochasztikusról. A dinamikus rendszerek lehetnek időben folytonosak, de állhatnak diszkrét időpontokból is.

Bevezetésként vegyük Ian Stewart kedvenc példáját: a vaddisznókat és a szarvasgombát. Ha sok szarvasgomba van, akkor nő a vaddisznók száma, ha kevés, akkor csökken. Ha sok a vaddisznó, akkor kevés gomba marad utánuk, míg ha csak néhány szaladgál, akkor elszaporodik a gomba. Ezeket a feltételeket leírhatnánk matematikai eszközökkel is, ekkor egy differenciálegyenlet-rendszerhez jutnánk. Bennünket az érdekel, hogy egy adott gomba- és vaddisznóállomány mellett hogyan alakul a számuk a jövőben. A



— Csak egy integrálszámító program van rajta...

differenciálegyenlet-rendszer megoldása nem olyan egyszerű feladat, s egy függvény mint megoldás első látásra nem mond túl sokat a rendszerről. Poincarénak jutott eszébe, hogy használjuk a geometriát és a képzelőerőnket. A változóink számának megfelelő dimenziójú ún. fázistérben rajzoljuk le, hogyan alakulnak idővel függvényeink értékei. Példánkban csak disznók és gombák szerepelnek, így kétdimenziós lesz a fázistér, tehát rajzunkhoz elegendő a sík is. E sík minden pontjához tartozik egy-egy görbe, amely megadja függvényeink alakulását. Lehet, hogy e görbék egy ponthoz tartanak, majd ott megállnak, lehet az is, hogy idővel visszatérnek a kiindulópontba, és az is megeshet, hogy egyik-másik görbe valamerre elmegy, és vissza sem jön. E görbeseregeket, amelyek lényegében megadják a folyamat viselkedését, attraktoroknak nevezzük.

Ha például egy vízcseppet ejtek a markomba, akkor az a vízcsepp valamilyen úton-módon eljut a tenyerem közepéig. Ha viszont megfordítom a kezem, és a kézfejemre cseppentem a vizet, akkor az onnan „menekül": leesik a kezemről valamelyik oldalon vagy az ujjamnál, vagy ráfolyik a karomra. Mivel mindenkinek elég változatos a kézfeje, a majdnem ugyanoda eső vízcseppek is teljesen más utakat választanak. Ezt precízebben úgy is megfogalmazhatnánk, hogy kicsit eltérő kezdeti feltételek lényegesen különböző végkifejlethez vezetnek.

A dinamikus rendszerekkel foglalkozók figyelmét ez utóbbi köti le, s nem véletlenül. Környezetünkben igen sok jelenségre (például az időjárásra) ez a tulajdonság teljesül. Valaha volt egy olyan metafizikai elképzelés, hogy ha pontosan ismerjük minden atom helyzetét és mozgásának irányát, akkor tetszőleges későbbi időpontban megadhatjuk az egész rendszer állapotát. Ezzel csak egy probléma van, és ez a „pontosan" jelentésével függ össze. Aki szokott kísérletezni, az tudja, hogy minden mérés hibával terhelt. A hibák kiküszöbölésére szoktunk több mérést végezni, és abban reménykedünk, hogy a hibák ellensúlyozzák, kiejtik egymást. (De azok a fránya hibák nem mindig tudják egymást kiejteni, hanem halmozódhatnak is az idők során.)

Nem nehéz feladat olyan rendszert kitalálni, amely ezeket a hibákat előhozza. Két ilyen rendszert elindíthatunk egymásétól csak a századik, vagy akár az ezredik tizedesjegyben eltérő kezdeti adatokkal, de hosszú párhuzamos működés után ezek is „szétmásznak". Például nyissunk ki két vízcsapot annyira, hogy éppen csak csepegjenek. Próbáljuk szinkronba hozni őket. Hamarosan látni fogjuk, hogy ha egy kis ideig együtt is csepegnek, pár perc múlva már biztos az összevisszaság.

Kaotikus viselkedés

Érdemes lenne utánajárni, hogy ki alkalmazta először (és terjesztette el) ezekre a jelenségekre a káosz kifejezést. Ezzel a „blikkfangos" megközelítéssel ugyan jó propagandát csinált ennek a tudománynak, de sajnos nem pontos a név. A determinisztikus dinamikus rendszerek elvileg pontosan előre jelezhetők a pontos kezdeti állapot ismeretében, rövid időt tekintve pedig a gyakorlatban viszonylag jól előre jelezhetők. A hosszú távú előrejelzést csak a kezdeti feltételektől való függés érzékenysége határozza meg. Ahhoz, hogy egy rendszer kaotikus legyen, igen nagy méretűnek kell lennie, és sok instabil állapottal kell rendelkeznie. Van egy többé-kevésbé szoros kapcsolat a káosz és a fraktálok között: a fraktál attraktorú dinamikus rendszerek kaotikusak. Bár ez alól mindkét irányban van kivétel, de első megközelítésben talán megfelelő.

Az érdeklődő olvasóknak javaslom, hogy írjanak egy aprócska kis programot, melyben az x := a * x * (1-x) viselkedését követik nyomon. Itt x-nek adunk egy tetszőleges kezdeti értéket, majd kiszámoljuk x újabb értékét a képlet alapján, és ezt folytatjuk szinte a végtelenségig. Az érdekes az, hogy az *a* megválasztásától hogyan függ sorozatunk viselkedése.

Ha valaki nem szereti a programokat, akkor kérje meg vegyész barátját, hogy mutasson erre egy példát. Van ugyanis néhány olyan anyag, melyeket megfelelő arányban összeöntve az elegy körülbelül hússzor periodikusan megváltoztatja a színét. (Ha még jól emlékszem fehér, narancs és kék színeket vesz fel sorra egymás után.)

Sejt és élet

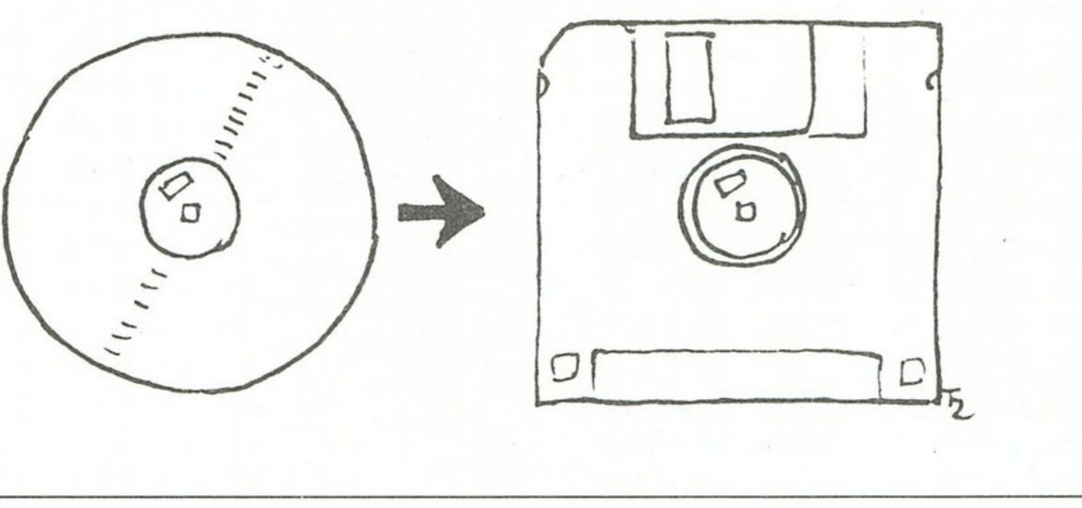
Aki a látványosabb programok iránt érdeklődik, az vegyen elő egy életjátékprogramot, mert az maga a káosz. A vizsgálatba bevont dinamikus rendszerek jellemzői kiválóan illenek a sejtautomatákra, így az életjátékra is. Ezért nem csoda, hogy a vaddisznó és gomba páros helyett a fű, nyúl és róka hármast már egy kellőképpen elbonyolított sejtautomatán vizsgálták. Az volt a meglepő, hogy az igen sok szabállyal megadott rendszer viselkedését nagyon jól lehetett modellezni egy négydimenziós fázistérrel is, tehát bonyolult problémáknak is lehet egyszerű szerkezetük.

Ez utóbbi azért is fontos, mert ha már vannak sejtéseink a dinamikus rendszerekkel kapcsolatban, akkor azt felhasználhatjuk a gyakorlatban is. Ha ismerjük egy rendszer jellemzőit, azaz az attraktorait, akkor egy adott ponton kis munkával sokat elérhetünk. A leghíresebb alkalmazás talán az volt, amikor hajtóanyaggal már alig rendelkező műholdat sikerült olyan pályára állítani, hogy azt korábban remélni sem merték. Csak annyit kellett tenni, hogy megfelelő helyeken és megfelelő időpontokban egy kicsit mindig odébb kellett rakni a műholdat, és kész. Lehet, hogy ilyen alkalmazások a Földön is elterjednek, és ezáltal talán még a mosógép is kevesebb áramot fog fogyasztani.

Az elmélyült kutatásoknak köszönhetően mind a fraktállal, mind a káosszal kapcsolatos irodalom bőséges. (Az Alaplap 1993. júniusi számában is ez volt a hónap témája.) Minthogy a mostani csak egy kedvcsináló cikk, a képleteket, pontos definíciókat megpróbáltam mellőzni. Akiket ez érdekel, az olvassa át a fraktál és káosz FAQ-okban felsorolt műveket.

Aszalós László

A kör (számítástechnikai) négyszögesítése



Ahol a félgyűrű nem fél gyűrű

A modellezés tudománya

A modellkészítés és a modellhasználat — röviden modellezés — nélkülözhetetlen fontosságú művelet az emberi élet minden területén és minden síkján. Az informálást is szolgálni képes ősi modellezés a beszéd és az írás is. A számok modellek, a számolás modellezés. Modellek a rajzok, a rajzolás pedig modellezési művelet. A matematika két ősi ága, az aritmetika és a geometria a számolásból, és a körzővel, vonalzóval dolgozó rajzszerkesztésből fejlődött ki. A matematika ma sem más, mint modellek tudománya.

A modellhasználat tudományos célja az információszerzés. (Vannak persze szórakoztató, illúziókeltő és más célú modellek is.) Tudományos modellhasználattal az információszerzés közvetett információszerzés. Ha közvetlen megfigyeléssel, méréssel lehetetlen, veszélyes, nehéz vagy túl költséges az információszerzés, akkor hívjuk segítségül a modelleket, és ezeken végzett mérésekből, megfigyelésekből következtetéssel állítjuk elő a kívánt információt.

Sokféle modell van

Az egyik modell erre alkalmasabb, a másik arra, az egyik használata olcsóbb, a másiké drágább. Az egyik több hasznot hoz, a másik kevesebbet. Ezeket és sok más szempontot körültekintően figyelembe kell venni, amikor modellekkel foglalkozunk, mert ezek határozzák meg egy modell, illetve egy konkrét modellhasználat értékét. Sajnos a modellezéssel eddig nem foglalkoztunk fontosságához illő tudományos alapossággal, az értékesség kérdése például még a szélesebb körben való tudatosulásig sem jutott el.

Életünk tele van modellekkel, sőt modellek modelljeivel. A puszta (tudományos) leírás, megjelenítés, reprezentáció is modell. A leírások, reprezentációk, megadásmódok fontossága rendkívül nagy. Gondoljunk csak a számok római típusú számábrázolási módszerére és a hindu típusú helyiértékes rendszerre! Amíg római számokat használtak Európában, addig majdnem mindenütt csak az összeadást, kivonást és a szorzást oktatták. Alig volt hely, ahol az osztást is tanították volna, annak nagyon nehéz(kes) volta miatt.

Napjaink érdekes kérdése a formulaírás. A klasszikus zárójeles és az annál újabb keletű zárójel nélküli formulaírás előnyeit és hátrányait minden számítástechnikus jól ismeri. Senki sem tudja azonban, hogy ezen a nagyon fontos területen milyen messze (vagy milyen közel) vagyunk a különböző szempontú optimumoktól. (Még a bírálati szempontok összegyűjtése sem volt eddig fontos senkinek...)

A modellhasználat gyakorlata

A matematika eleve modellező praxist folytatott, egyetlen célja volt a modellezéssel való problémamegoldás, törekedett a minél több helyen alkalmazható modellek megalkotására, és a minél előnyösebben használható modellek megtalálására.

Kettős munka folyt. A külső világ problémáinak megoldása, a külső világ matematikai megmagyarázása, és a saját eszközök, módszerek tökéletesítése, minél több és különfélébb probléma felvetésén és megoldásán próbára téve az eszközök arzenálját. Nagyon jellemző a matematikára, hogy már a kezdetektől gyakorlat volt az önellenőrzés, a belső tisztaságra törekvés. (Ez mára teljesen háttérbe szorult.)

A konkrét problémák megoldásában a siker nélkülözhetetlen kelléke *a modell megalkotni tudása* és *a modell kezelni tudása.* Válaszolnunk kell olyan kérdésekre, mint például a következők:

— Mit modellezünk? (Helyzetet? Folyamatot?)

— A modell milyen igényt elégít ki? És azt milyen mértékben elégíti ki?

— Mivel a modellek általában absztrakció eredményeképpen jönnek létre, pontosan tudnunk kell azt is, hogy a tapasztalati világból absztrahálunk, vagy már egy előbbi absztrakció eredményéből.

A matematika lényege a modellezés, a modell pedig minden esetben egy kis világ, saját belső törvényekkel, tulajdonságokkal. A matematika természetéből eredően termeli a modelleket, tehát alkotja a kis világokat. (A modell helyett a következőkben a *kisvilág* kifejezést is használni fogjuk.) Idővel a hasznosnak bizonyult modellekből modell(kis)világok egész tömege jött létre. Ezekből sokféleképpen sokféle információ nyerhető. Minden esetben kulcskérdés, hogyan jutunk a kívánt információ birtokába. Az egyszerűbb esetekben ez annak a megállapítását jelenti, hogy a modellezett és a modell viszonyában minek mit feleltetünk meg.

A modell a modellezettnek valamilyen leírása. Lényegi kérdés, hogy mikor mit, mivel modellezünk. Ha matematikai objektumokkal, rendszerekkel, kisvilágokkal modellezünk valamit, akkor az matematikai modellezés. Általában az igaz, hogy ha az információnyerésben bármilyen matematikai művelet lényeges szerepet játszik, akkor matematikai modellezésről beszélhetünk.

A matematikában a modellezéshez eszközként bizonyos objektumok és viszonyok leírásának és kezelésének lehetősége áll rendelkezésre. A jelenleg kínált lehetőségek lényegében statikusak. Az időbeliség igazi megragadása a matematikának még nem sikerült. A papír-ceruzás „soros" gondolkodásmód és (modell)világ nem alkalmas a való világ legtöbb jelenségének a kezelésére, még akkor sem, ha nem papírral, ceruzával, hanem számítógéppel dolgozunk. Lehet, hogy új modellező eszközként mégis a számítógép lesz az, melynek révén megugrik a matematika fejlődése. (A sok párhuzamosan dolgozó processzor és a sok párhuzamosan dolgozó gép fogja kikényszeríteni az új gondolkodásmódot és az új eszközkészlet megalkotását.)

A matematika kisvilágai

A matematika kisvilágai, bevált modelljei — hitünk szerint — absztrakcióval jöttek létre. Az absztrakció pedig egyszerre áldás is és átok is. Átok akkor, amikor szokássá válik. Különösen súlyos átok, ha absztrakciót végzünk minden más művelet helyett, például a konkretizálás helyett is. Korunk matematikája a minden helyett absztrahálás

átka alatt vergődik. Az absztrakcióval megtermelt „produktumok" túltermelési válságát sínyli a jelenkori matematika. Vannak persze régi, jól bevált absztrakciós produktumok is. Vessünk ezekre egy pillantást!

A számok összessége, a sík pontjainak összessége, bizonyos tulajdonságú függvények összessége stb. kisvilágok, saját tulajdonságokkal, saját törvényekkel. Ezek mint modellek a nagy világ, a teljes világ részeinek, ezek jelenségeinek, helyzeteinek, folyamatainak modelljei. A kisvilágok tulajdonképpen világvetületek, világtranszformációk, amelyekből felépítjük a modellezett világ helyzeteinek, folyamatainak megismerési eljárásait.

A kisvilágokat a matematika azok belső szerkezetével, struktúrájával jellemzi. Hivatalosan háromfajta alapstruktúrát különböztetünk meg: a kisvilág algebrai, rendezési és topológiai struktúráját.

A kisvilág algebrai struktúrája lehet gruppoid, félcsoport, csoport, félgyűrű, gyűrű, ferdetest, test stb. Ezek a szavak különböző törvényű hozzátartozási jelenségek meglétét vagy hiányát hivatottak kifejezni. (Bármely két egész számhoz pl. hozzátartozik azok összege és szorzata, amely szintén egész szám, és érvényes rájuk az ún. „disztributivitás" „törvénye".)

A kisvilág rendezési struktúrája arról informál, hogy mely elemei melyeket előznek meg.

A kisvilág topológiai struktúrája ismeretében a „folytonossági" kérdésekben tudunk állást foglalni.

Természetesen nem minden kisvilágnak van algebrai, rendezési és topológiai struktúrája. Van olyan kisvilág például, amelynek csak egyféle struktúrája van. (Elgondolkodtató, hogy a mai matematika a geometriával, a számítástechnikával és a párhuzamos működésű — egymást vezérlő — rendszerekkel nem tud mit kezdeni. Ezekről igyekszik tudomást sem venni.)

Érdekes lenne végigelemezni a matematika tényleges szerepét a mai életben, de be kell érnünk néhány gondolatébresztő kérdés felvetésével.

A matematika tényleges szerepe

Milyen a matematika és a számítástechnika viszonya?

Mi a számítógép szerepe a matematikai kutatásokban? (Ismert tény, hogy a híres „négyszínsejtés" bizonyítása gépi úton történt.)

Mi a matematika szerepe a számítástechnikában?

Mi a matematika szerepe az életben? Az élet „computerizálódása" révén mennyire matematizálódnak a folyamatok? A számítástechnika terjedésével a matematikai egzaktság is terjed, vagy a számítástechnika matematikán belüli terjedésével a matematikában is terjed a számítástechnika jellegzetes amerikai trehánysága?

Reménykedhetünk-e a matematika és a számítástechnika együttes oktatásában, vagy a tantárgyi sovinizmus el fogja taposni a józan észt?

Mit hoz a jövő, ha a matematika megőrzi hagyományosan jó elriasztó eszközeit? A régi görögök vallották, hogy az istenek aritmetizálnak. Egy jelenkori matematikatörténész szerint a matematika istentisztelet. Igaz mindkét vélemény. De az is igaz, hogy a józan eszű ember falra mászik a következő matematikai igazságoktól:

— Nem minden ferde test ferdetest.

— A ferde hasáb például nem ferdetest, hanem ferde test.

— Minden test ferdetest, de nem minden ferdetest test.

— A félcsoport nem fél csoport, hanem félig csoport.

Fejlődés, változás, erózió

Fejlődik-e, változik-e a matematika? Tartalmában, eszközeiben, szemléletében? Terminológiájában biztosan nem. Minden komolyabb tudománynak van rendes, logikus nomenklatúrája, nevezéktana. A matematikának nincs. (A matematikában bármely buggyant agyú alkothat szakkifejezéseket, az igazi nagy matematikusok pedig olyan szakkifejezéseket alkotnak, hogy az biztosan kiváltsa a buggyant agyúak tetszését.)

Az ötvenes évektől úgy látszott, hogy az „operációkutatás", nevezhetjük optimumtudománynak is, átformálja és megújítja a matematikát. Ma már nyilvánvaló, hogy ez nem következett be. Olyannyira nem, hogy az operációkutatás napjainkra szinte teljesen eltűnt a matematikából.

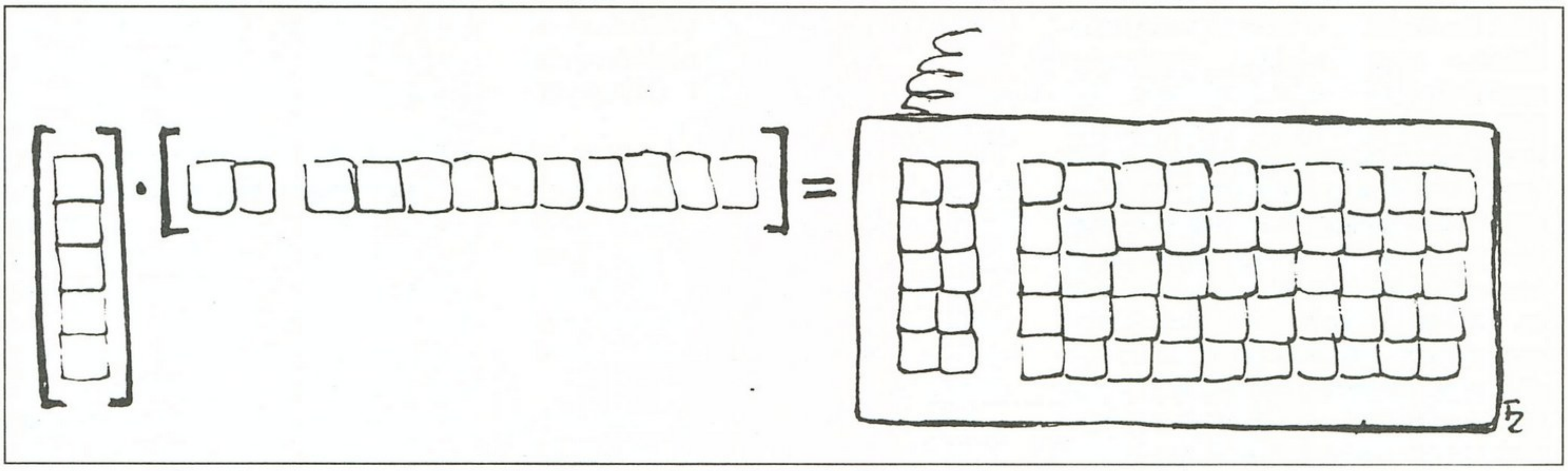
Reménykedtünk, hogy a számítástechnika révén terjedni fog a matematikai egzaktság is, legalábbis a logikus gondolkodás. Nem ez történt. Olyannyira nem, hogy a matematikai szigorúságú logikának az ún. logisztikának a nevét is elbirtokolták, és másra használják. Nagyon jellemző, hogy mire. Szállítási operációkutatásra. Tehát olyanoké a világ, akik újra felfedezik az operációkutatást, és a matematikai logika nevét adják neki.

A matematika eróziója újabban a tankönyvekben is megmutatkozik. Az általános iskolától az egyetemekig egyre több az olyan matematika-tankönyv, amely hemzseg a durva hibáktól.

Ma nyugodtan kaphat tudományos fokozatot az is, aki nem tudja, mi az ekvivalencia reláció, és nyugodtan lehetnek egyetemi tanárok, akik szintén nem tudják ezt vagy a matematikai analízis legfontosabb tételeit, és ezt nem is tartják hibának.

A jelekből ítélve úgy tűnik, hogy a matematika fejlődése fordulóponthoz érkezett. A jövő attól fog függni, hogy szerepet kap-e, és mekkora szerepet kap a matematikában az erkölcs. Erkölcsi kérdés ugyanis, hogy egy tudomány művelői embertársaik javán fáradoznak-e, vagy saját szórakoztatásukat űzve, eltartatják magukat a társadalommal.

Pogány Csaba



Felkutatni, amit kikutattak

Szakcikkek közt keresgélve

Érdemes figyelemmel kísérni a mások által elért dolgokat, mert egyébként könnyen előfordulhat, hogy olyasmire fecséreljük időnket, amit mások már előttünk kidolgoztak, megalkottak, felfedeztek... Egy adott témakörrel foglalkozók nagy száma miatt arra nincs lehetőség, hogy mindig mindenkivel személyre szóló levélben tudassuk új eredményeinket. Ezért találták ki a folyóiratokat, a konferenciákat. Az alábbi eszmefuttatás konkrétan a matematikát ragadja ki a sokféle szakterületből, de más szellemi tevékenységet folytatók számára is utat mutat, hiszen sok az azonosság ezekben a helyzetekben.

Mire is használhatják a matematika művelői a számítógépeket? A matematika egészét a könnyebb eligazodás kedvéért diszciplínákra bonthatjuk. A megszokott analízis, algebra, geometria és valószínűségszámítás felosztást persze tovább is lehet finomítani, nagyjából háromezer ágra. Ezek az ágak gyakran egymástól függetlenül fejlődnek, majd hirtelen valaki felfedezi, hogy milyen jól lehet az egyik helyen bevált módszereket egy más helyen is felhasználni, és ekkor e két terület rövid időre összekapcsolódik, együtt fejlődik. Például a számelmélet (amely legközelebb az algebrához áll) előszeretettel használja eszközül a komplex függvények elméletét. Ha véletlenül felfedezzük, hogy az éppen megoldandó probléma egy másik tudományterülethez is tartozhat, akkor gyorsan szét kellene nézni, hogy nincs-e ott kész megoldás. Saját területét persze mindenki jobban ismeri, tudja, melyek a bevezető művek, hol található rá színvonalas publikációkra. Ez más, esetleg egészen fiatal tudományterületekre általában nem jellemző.

Pár száz évvel ezelőtt az ismert matematika csak töredéke volt a mainak, és a matematikával foglalkozók átláthatták az egész tudományágat. Még arra is volt idejük, hogy más tudományokban is maradandót alkossanak. Akkoriban nem foglalkoztak nagyon sokan a tudományokkal, és a tudósok közötti kapcsolattartásra elegendő volt a levelezés, hisz csak néhány embernek kellett bejelenteni az új felfedezéseket.

Mára ez gyökeresen megváltozott. Nagyságrendileg több ember foglalkozik ezzel a tudománnyal (is). Ezért a matematika tudásanyaga a kívülálló számára felfoghatatlan mértékben nő. Hatalmas könyvtárakat tölt meg, és a fejlődés egyre csak gyorsul. Ahogy nőtt a tudósok száma, úgy gyarapodtak a folyóiratok és konferenciák is. Eljutottunk oda, hogy lehetetlen részt venni az összes érdekes konferencián, mert sem idő, sem pénz nincs rá, illetve annyi a folyóirat, hogy a könyvtárak nem képesek az összeset járatni, vagy ha járatnák is, senki nem lenne képes mindet átlapozni.

Csak szűkebb adatok

Az eligazodás megkönnyítésére találták ki azokat a kiadványokat, amelyekben csak azt sorolják fel, hogy mi és hol jelent meg. Már ezek a könyvek is hosszú polcokat töltenek meg a könyvtárakban, és bennük sem olyan egyszerű ráakadni a kívánt információra, noha igen alapos munkával készítették el azokat. Könnyű ugyanis olyan kérdéseket feltenni, amelyekre szinte lehetetlen megtalálni a választ, pedig az valahol ott lapul. Szerencsére a könyv kiadója haladt a korral, és megjelentette az egészet CD-n is. Mivel egy CD-re minden nem fért fel, egy átlagos számítógépen kényelmetlen használni, hiszen állandóan cserélgetni kell a lemezeket.

Az egész adatbázis szöveges, a megjelent cikkek azonosítóit tartalmazza (szerzők, megjelenés éve, folyóirat címe és száma), valamint azt, hogy a cikk mely osztályba sorolható be, mi a rövid tartalma, ki nézte át megjelenés előtt stb. Igen gazdag a lekérdező nyelv, kereshetünk szavakra, szókombinációkra, szűkíthetjük a keresést azzal, hogy ezek a szavak mely mezőkben szerepeljenek, és természetesen használhatjuk ezek logikai kombinációit. Mivel az utóbbi ötven év termésének java megtalálható itt, igen okosan kell megfogalmazni a kérdést, hogy ne több ezer cikkre vonatkozzon. Miután a keresés végetért (azaz pár másodperc múlva), sorra elolvashatjuk az adott cikkekre vonatkozó adatokat, az érdekesebbeket kijelölhetjük, és lementhetjük lemezeinkre a későbbi feldolgozáshoz.

Hivatkozások

Az előbb felsorolt információk általában a leglényegesebbek, de engem gyakran más is érdekel. A matematikusokra nem az jellemző, hogy az alapoktól kezdenek mindent, hanem már valami készre építkeznek. A cikkeik végén felsorolják, mely cikkekre támaszkodtak, azaz mely cikkek állításait, tételeit, definícióit használták fel. Azt nem nehéz összegyűjteni, hogy egy adott cikk mire hivatkozik, azt viszont már igen, hogy az adott cikkre később ki hivatkozott és hol.

Ha valahogy már összegyűltek ezek az adatok, akkor statikus adathordozókon kényelmetlen a használatuk abban az esetben, ha az újabb verziók csak a bővítéseket tartalmazzák. A másik probléma ezzel a CD-s módszerrel az idő. A folyóiratoknak csak a már megjelent számaira vonatkozik, és az adatok azokról is bizonyos késéssel kerülhetnek be az adatbázisba, így nem juthatunk hozzá azonnal a legfrissebb információkhoz. Ráadásul ezek az adatbázisok nem tartalmazzák a világon megjelent összes kiadvány minden adatát, csak a nívósabbakét. Ezeken a problémákon — hasonlóan sok más esethez — az Internet segíthet.

Megoldás lehetne például az a most elterjedt módszer, hogy a CD-n megjelent anyagokat kiegészítő friss információt bizonyos időközönként le lehet tölteni. Rendszeres használat esetén ez azonban elég sokba kerülne. Ennél szerintem egy fokkal jobb az a módszer, ha csak a számunkra értékes információt töltjük le. Ekkor szinte mindegy, hogy az adatbázis mekkora része található meg nálunk, lehetne akár az egész is egy távoli gépen.

Számomra a logikával kapcsolatos témák érdekesek, és egy ehhez kapcsolódó adatbázis van a http://info.gte.com/dblp/db címen. Itt lényegében

több tucat folyóirat anyaga lett összegyűjtve és több szempont szerint rendezve. Az egyik ilyen szempont, hogy az újság egy adott számában mi jelent meg. Ennél jóval érdekesebb az, hogy egy adott személytől mi és hol jelent meg. Ha ennek az embernek weblapja is van, akkor jobb esetben azt is feltünteti az adatbázis. A cikkek mellett gyakran egy rövid kivonatot találhatunk, amelyből eldönthetjük, hogy érdemesnek látjuk-e elolvasni a cikket. A cikk ismérveit letölthetjük, és könnyedén beemelhetjük hivatkozásként.

Ezek voltak az adatbázisok. Léteznek viszont olyan helyek is, ahol nem a cikkekre vonatkozó adatok, illetve azok kapcsolatai és keresési lehetőségei állnak a központban, hanem maguk a cikkek. Minden magára valamit is adó oktatási intézmény (így például a KLTE is) a weblapjai közé felvesz olyat is (például a http://www.math.klte.hu/publi), amelyen az ott tanítók cikkeit, eredményeit bemutatják. Ez lenne a legjobb megoldás: ott tárolni az információt, ahol létrehozták! De mindenütt mások a szokások.

Vannak olyan helyek, ahol egységesen kezelik az összes cikket, és vagy a dátuma, vagy a szerzője szerint csoportosítva férhetünk hozzájuk. Azután vannak olyan helyek, ahol rábízzák a szerzőre, hogy a saját weblapján úgy jelentesse meg a cikkeit, ahogy akarja. Ezzel az a baj, hogy a szakma nagy öregjei, akik meghatározó személyiségei az adott tudományágnak, nem használják úgy és annyira a számítógépeket, mint a fiatalok. Magyarán inkább az ifjabb korosztály készíti el saját bemutatkozó lapjait, és helyezi el a Weben a kutatásaival kapcsolatos anyagait.

TeX-állományok sorsa

Mióta Knuth elkészítette a TeX rendszert, egyre több folyóirat kéri arra a szerzőit, hogy a szerkesztőség által egységesített stílust használva készítsék el TeX-állományaikat, és ezt küldjék el e-mailben a szerkesztőségbe. Ezzel a szerkesztőség munkája leegyszerűsödött, nem kell a cikket újra legépeltetni, majd ezt a változatot egyeztetni az eredetivel, a hibákat kijavítani. (Ezért, ha valami elírás akad a szövegben, az biztos, hogy a szerző keze nyomát dicséri.) Továbbá az elektromos levél gyorsabban és olcsóbban jut el a címzetthez, mint hagyományos társa. Tehát felgyorsulhat a kapcsolattartás, és a cikk is gyorsabban megjelenhet.

Ezen kényszerítő okokból egyre több matematikus (még az idősebb korosztályból is) kezdi használni a TeX-et, és ezért a számítógépeket. Ha ugyanis egy cikk elektronikus formában már létezik, akkor semmibe sem kerül azt elhelyezni egy ftp-szerveren vagy egy weboldalon. Léteznek olyan gépek is az Interneten, amelyek cikkszerverként működnek. Hivatalosan fájlszerverek, s így a levélben vagy ftp-n keresztül érkező kéréseket teljesítik, de mivel itt csak cikkeket tartalmazó TeX-fájlok találhatók, szerintem nyugodtan alkalmazható rájuk a cikkszerver elnevezés is.

Vannak, akik túlzásnak tekintik cikkeik forrásának közkézre adását, és csak DVI vagy PostScript verzióban publikálnak az Interneten. Ebben az esetben én a DVI híve vagyok, mivel tömör formátumú, szemben a bőbeszédű PS-szel, és sokkal jobb nézegetők is léteznek hozzá. De ha egy cikk tele van mindenfajta ábrával, grafikával, akkor tényleg a PS a legegyszerűbb megoldás. Napjainkban kezd divatossá válni az Adobe Acrobat (pdf) és az SGML formátum használata is, bár szerintem nehéz túllicitálni a TeX-et, főleg a matematikai képletek megszerkesztésében. Jelenleg ezért is próbálkoznak egyesek a tex2pdf átalakítókkal, nem tudom, milyen sikerrel.

Véleményem szerint a Weben való publikálást hátráltatja, hogy a HTML szabvány még nem kiforrott. Egyre jönnek az újabb változatok, velük együtt a nézegetők újabb generációi. A Netscape és az Explorer egymást kergeti, és megjelentek a dobozok (frame) is. Mielőtt valamelyik cég végleg feladná a küzdelmet, a karakteres nézegetőket használók már elvesztették a harcot, hiszen a bedobozolt oldalakat azokkal nem lehet megjeleníteni. Ha valamelyest már lecsillapodik a WWW-őrület, a HTML vagy valamely utóda sem változik napról napra, továbbá lesznek könnyedén használható WWW-szerkesztők, akkor talán megjelennek az ebben a formátumban megírt matematikai cikkek is.

Keresőgépekkel

Hogyan tudunk ráakadni az Interneten arra az oldalra, ahonnan letölthetjük a számunkra érdekes cikkeket? Bár a helyzet hasonló, de szerencsére nem olyan reménytelen, mint a tű megtalálása a szénakazalban, mert használhatjuk a keresőgépeket. Személy szerint én a Yahoot és az AltaVistát használom. Gyakran legfeljebb egy-két olyan oldal létezik, amelyre éppen kíváncsiak vagyunk, például, hogy merre van S. E. Nki-nek a weblapja (ha egyáltalán van neki).

Kacifántosabb a dolog, ha a keresett személynek Kovács János-szerű neve van. Ilyenkor jól jön, ha elolvassuk a keresőprogram használati utasítását, különösen a bonyolultabb keresésekre vonatkozó fejezetet. Megfelelő logikai szűréssel elkerülhetjük, hogy a keresőprogram több ezer találattal traktáljon bennünket, és azokat manuálisan kelljen végignézni.

Az AltaVista kritériumai például nagyon egyszerűek, egyrészt idézőjelek között szerepeltethetünk összetartozó szókapcsolatokat, mint például neveket, kifejezéseket. Aztán használhatjuk az ÉS, VAGY, NEM, illetve KÖZEL logikai kifejezéseket is (persze csak angolul).

Tegyük fel, hogy már van egy szerzőnk, és esetleg egy cikkcímünk is. Érdemes először is megnézni, hogy van-e az illetőnek weblapja (keressünk rá a teljes nevére), mert ott általában mindenki elhelyezi a cikkeit is. Ha ez nincs, de tudjuk, hogy hol dolgozik, el lehet indulni az adott intézmény weblapjáról is, hátha szerencsénk lesz. Ha ez sem vezet eredményre, akkor valamelyik adatbázisban kell utánanézni, hogy kivel szokott közös cikkeket megjelentetni. Lehet, hogy a kollégája már rendelkezik saját oldalakkal, és ott helyezte el ezt a cikket is. Ha nagyon fontos a cikk, esetleg írhatunk magának a szerzőnek is. Nagy valószínűséggel válaszolni fog, hiszen mindenkinek hízeleg, ha a világ másik végéről is érdeklődnek a cikkei iránt.

Aszalós László

DÁTUMOK • UP TO DATE

SCHALL SZAKVÁSÁROK

SCHALL vásárok a piac igényei szerint!

DÁTUMOK • UP TO DATE

	SCHALL SZAKVÁSÁROK		1997	1998/99
QualiFinn	A minőségbiztosítás nemzetközi szakvására **Lahti, Finnország**	**1996: 158 kiállító 2673 látogató 4000 m² brutto kiállítási terület**	2. QualiFinn **1997. augusztus 27-29.**	——
Motek	Szerelési- és karbantartástechnikai nemzetközi szakvásár **Sinsheim, Németország** Újdonság: Demotec bontástechnológiai szakkiállítás	**1996: 584 kiállító 19 185 látogató 24 500 m² brutto kiállítási terület**	16. Motek **1997. szeptember 23-26.**	17. Motek **1998. szeptember 22-25.**
Fakuma	A műanyagfeldolgozás nemzetközi szakvására **Fridrichshafen, Németország**	**1996: 1072 kiállító 30 430 látogató 50 000 m² brutto kiállítási terület**	12. Fakuma **1997. október 14-18.**	——
AW-L Tech	Szennyvíz- és víztechnológiai szakvásár **Sinsheim, Németország**	**1996: 70 kiállító 1 200 látogató 4 500 m² brutto kiállítási terület**	4. AWL-Tech **1997. október 22-24.**	——
Qualitec	A minőségbiztosítás nemzetközi szakvására Elmia, Jönköpingi **Vásárközpont, Svédország**	——	——	1. Qualitec **1998. március 24-26.**
Südblech	Lemezmegmunkálási és illesztéstechnikai szakvásár **Sinsheim, Németország**	**1995: 139 kiállító 4 092 látogató 8 000 m² brutto kiállítási terület**	——	3. Südblech **1998. március 24-27.**
Fameta	Fémmegmunkálási nemzetközi szakvásár **Nürnberg, Németország**	**1996: 632 kiállító 29 818 látogató 28 400 m² brutto kiállítási terület**	——	16. Fameta **1998. május 5-9.**
Control	A minőségbiztosítás nemzetközi szakvására **Sinsheim, Németország**	**1997: 618 kiállító 16 836 látogató 26 000 m² brutto kiállítási terület**	——	12. Control **1998. május 12-15.**
Optatec	Optikai, optoelektronikai nemzetközi szakvásár- Felhasználás és technológia **Frankfurt/Main, Németország**	**1996: 264 kiállító 3 919 látogató 9 600 m² brutto kiállítási terület**	——	4. Optatec **1998. június 16-19.**
Techmo	Szerelési- és karbantartástechnikai nemzetközi szakvásár Westalenhalleni Vásárközpont, **Dortmund, Németország**	**1996: 133 kiállító 2673 látogató 5 200 m² brutto kiállítási terület**	——	7. Techmo **1999**

Szervező:
P.E. SCHALL GMBH MESSEUNTERNEHMEN
Gustav-Werner-Strasse 6, D-72636 Frickenhausen
Tel: 00 49/70 25/92 06-0
Fax: 00 49/70 25/92 06-20

FAMA

Magyarországi képviselet:
OPEN GATES HUNGARY KFT.
1123 Budapest, Nagyenyed u. 6.
Tel/Fax: 1562-890
Tel: 214-9943, 214-9059
E-mail: OGH. Kft @mail.datanet.hu

DÁTUMOK • UP TO DATE

INFORMÁCIÓKÉRÉS: 0926 ▼



INFORMÁCIÓKÉRÉS: 0909 ▲



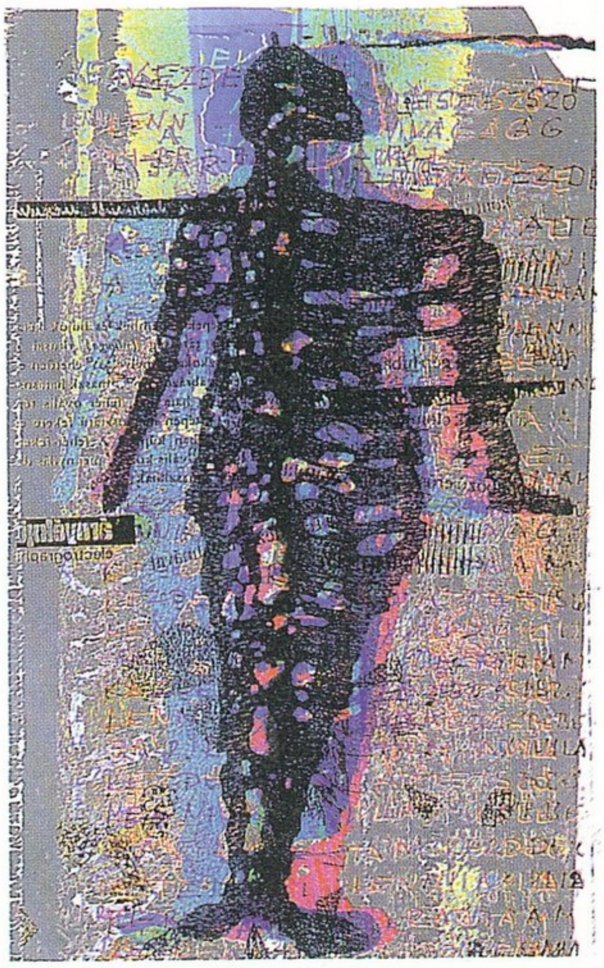
INFORMÁCIÓKÉRÉS: 0924 ▲



INFORMÁCIÓKÉRÉS: 0921 ▲

Magyar díjnyertesek

Számítógépgrafika

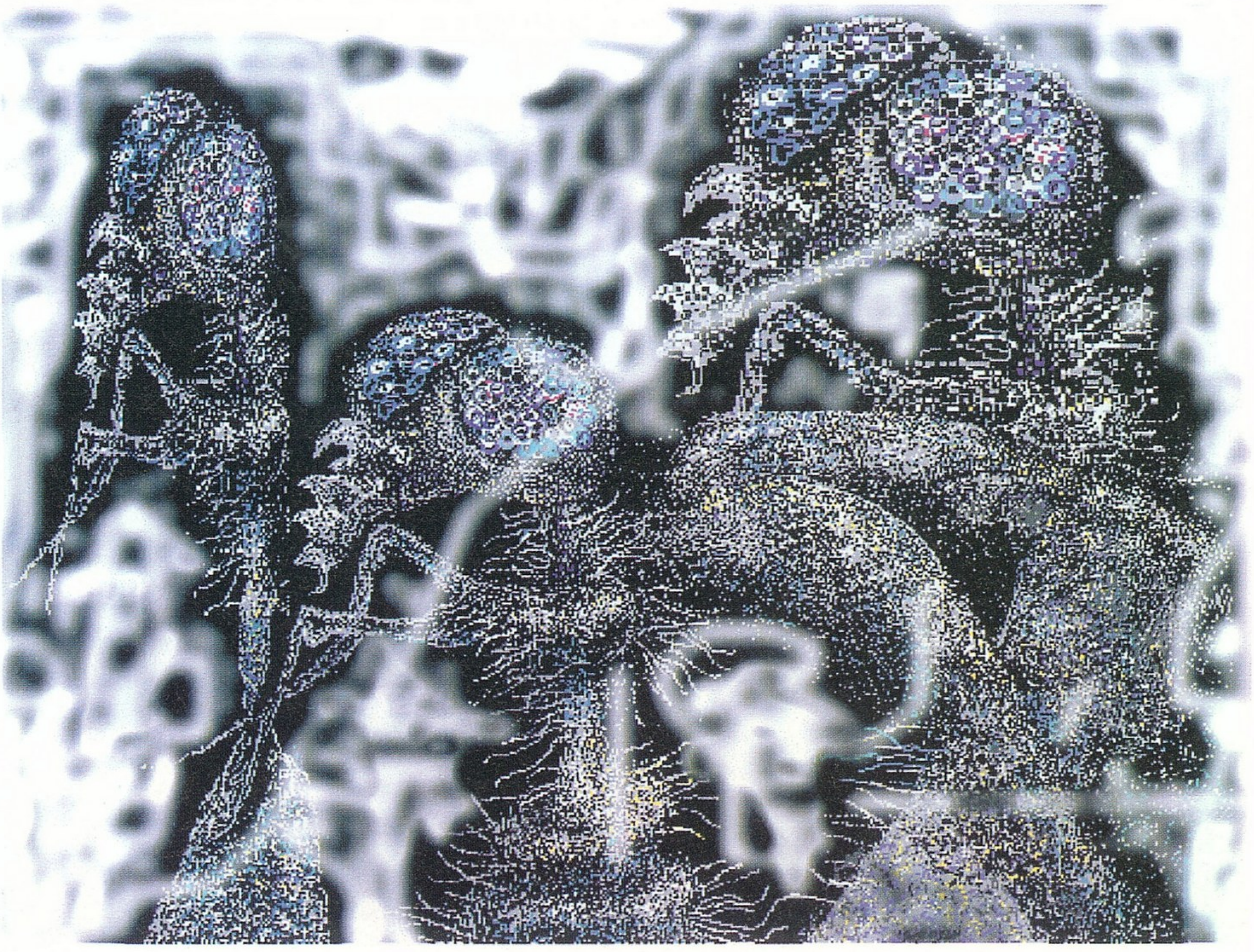


Vannak a tágabban értelmezett szakmának olyan eseményei, amelyekről a sajtó gyakorta csak áttételesen értesül. Most egyik külső munkatársunk hívta fel figyelmünket arra a rangos nemzetközi eseményre, amely annál is inkább megérdemli a publicitást, mivel kiemelkedő magyar sikerekkel zárult.

A nem-tradicionális és avantgarde grafika európai művelői június 26. és július 20. között Prágában adtak randevút egymásnak. A triennálén bemutatkozott alkotások mintegy 60%-a (!) készült számítógépen, s ez a műfajon belül a technika nagyfokú előretörését mutatja.

A chicagói Roman Berka által vezetett zsüri a triennálé első díját a cseh Pavel Cernynek ítélte oda, a magyar résztvevők közül Haász Ágnes ezüst-, Geller István bronzfokozatú elismerésben részesült, míg Hernádi Paula a műkritikusok díját érdemelte ki.

Alsó képünkön Hernádi Paula: Légy című alkotása látható, amelynek külön érdekessége, hogy az itt látható finom részletek kidolgozására az egér volt az egyetlen eszköz... Másik illusztrációnk Haász Ágnes: Kezdek áttetsző lenni című grafikája.



Kritikai vitriológia

Egy „eseménytelen” nyár

**Július 27-ét írunk, amikor én ezt most írom....
Mit tartogat még a nyár? Nehéz jósolni. Számítástechnikai szempontból a nyár egyébként is elég eseménytelenül zajlott. Az egyetlen fajsúlyosabb új program a Netscape Communicator, ezt viszont csak később kapom kézhez (mármint a végleges, dobozos változatot), így ez is kimarad az e havi vitriolból. Mellesleg nem szeretek félkész szoftverekről írni. Már tavaly télen is volt letölthető Communicator-kezdemény, de nem volt kedvem a bogarakat kerülgetni. Pláne nem rájuk taposni. Ez alól az elv alól kellemes kivétel a FAR Manager, amely stabil bétaként kezdte meg hívei fokozatos meghódítását.**

Szebben illik elbúcsúzni

Mennyi szépet és jót írtam annak idején a Quarterdeck cég memóriamenedzseréről, a QEMM-ről. Azt terveztem, ebben a számban bemutatom a QEMM 97-et, verziószám szerint a 9.0-s változatot. De ez inkább halottkémi jelentés lesz. A termék feleslegessé vált, és számomra gyakorlatilag kihunyt. Élt vagy 10 évet, de eljárt felette az idő. A Microsoft rendszerprogramozói megtanulták a memória hatékonyabb kihasználását. Ezt a QEMM nehezen akarja felfogni. A QEMM 97 a Windows 95 számára már csak azt az egyet ígérte, hogy a 32 bites alkalmazásokat 30–65 százalékkal gyorsabban tölti be a memóriába (kérdezem én: na és?!). Vagy nem. Az uninstallálása is meglehetősen kalandos volt, még a Windows 95 újratelepítése után is visszaköszöntéskellemetlenkedett.Össze kellett szednem minden eddigi rutinomat, hogy végleg (egyszer és mindenkorra) levakarjam a gépemről. Saját Uninstall modulja núdli. Álljon ez figyelmeztetésként mindazok számára, akiknek a QEMM évek óta a briliáns szoftvertechnikai megoldásokat jelentette.

A Quarterdeck egyébként még idejében felfogta, hogy erről a pályáról kitiltották, ezért fékevészett bevásárlásba menekült (előre), így ő birtokolja a legtöbb uninstaller programot (Cleansweep, Remove-It), bekebelezte a DataStormot (Procomm Plus család, végre NT-re is elkészült a 4.5-ös változat). Az internetes divízió szintén gyakorta mellényúl, de kérem, itt a túlélésről van szó. A QEMM hattyúdala nem sikeredett felejthetetlenné. Máris többet írtam róla, mint amennyit ez a fércmű megérdemelt volna. Kár érte. Az igazi művész a csúcson tudja abbahagyni. A Quarterdeck bizton több ellenséget és haragost szerez így, mintha hagyta volna a QEMM 8.0-át szép csendben kimúlni, elenyészni.

Internet-biztonság

Néhány gondolat és megjegyzés az Internet biztonságosságának kolosszális üzletét meglovagoló cégek hozzáállásáról. A TCP/IP protokoll legnagyobb hátránya éppen egyszerűségében rejlik. Ha nem választjuk el saját hálózatunkat egy ún. tűzfallal (firewall) a külvilágtól, bárki — jó vagy rossz szándékú — behatolhat a nyilvános IP címünkön keresztül gépünkbe. Fájlszinten. Ha csak nézelődne, még nem is lenne nagyobb baj. De fájlműveleteket is végre tud hajtani. Képzeljük csak el, mi lenne a banktitokkal, ha XY bank szakemberei be tudnának hatolni egy másik pénzintézet titkos adataiba. Bankrablás. Információrablás.

Egy hálózat elválasztása a külvilágtól, magától az Internettől nem egyszerű feladat. Lépést kell tartani az egyre okosodó hackerekkel, akiket ma már nyugodtan nevezhetünk ipari kémeknek is. Napjaink kémei vajon már PC-n szereznek hírt? És azon keresztül is igyekeznek elhárítani az ellenkémeket? Minden meglehet. Még az is, hogy egy kémet csőbe húznak. Kémcsőbe. Egy szó mint száz, a biztonság, az illetékesség nagy biznisz. Authenticity, secure computing, ezek az új bűvszavak. Azonban az ilyen szoftverek elkészítése egy amerikai szakíró szemléletes hasonlatával élve olyan, mintha egy robogó vonatot kellene kívülről átfestenünk.

Praxisunkban előfordult már, hogy neves ügyfelünk ilyen jellegű szoftvert vásárolt. Egy hónapjába került, amíg életet lehelt a nem kevés pénzéért kapott programkódhalmazba. Az elindításhoz többször kapott mind telefonon, mind pedig e-mailen keresztül téves indítókódot a fejlesztőktől. Amit eredetileg vett, az nem volt a gyári CD-n, hetekig hitegették, hogy mégis rajta van, majd hogy az új változatban ingyen megkapja, legvégül hogy ingyen letöltheti valahonnan a Webről. Ez végül nagy nehezen bejött. Azóta biztonságosan és boldogan élnek, amíg ki nem jön a szoftver újabb változata. Akkor minden kezdődik elölről.

Öt vagy több

Ez egy ötletes játék címe, magyarra fordítva az 5-Or-More! Az amőba egy jópofa kiterjesztése. A gép ellen kell játszani, és valóban az a cél, hogy minél többször egy vonalban (akár átlósan is) kijöjjön azonos színű jelekből egy sor. Ekkor a sor eltűnik a tábláról. Mellesleg a gép minden egyes lépésnél három új színes jelet dob fel véletlenszerűen a kilencszer kilences négyzet alakú táblára. Egy tetszés szerinti figurát megnyomunk, az villogni kezd, majd pedig megadjuk a végcélt. Amennyiben a jel útja nincs eltorlaszolva a kijelölt hely megközelítése irányába, úgy a gép áthelyezi a jelet a megadott helyre. Igyekeznünk kell minél hamarabb ötösöket, sőt hatosokat és heteseket létrehozni, mert igencsak gyorsan megtelik a tábla, és a jelek nem kevés szlalomozással tudnak csak közlekedni.

Nagyjából ennyi a szabály. Ez a játék is egyszerű, és nagyszerűsége éppen ebben rejlik. Nem az idő ellen játszunk, módunkban áll mindent komótosan megfontolni, mert csak a pontszámra megy a játék. Érdemes kipróbálni, noha valószínű, hogy a játék nem fogja elérni azt a sikert, amelyet annak idején a Tetris méltán elért a PC-s berkekben. Sajnos a kipróbálási változat nem enged 600 pont fölé menni. Kár, mert még tudtam volna rekordot rekordra halmozni. A programot egyébként egy kis

cég alkotta, amely eddig kisebb windowsos üzleti alkalmazásocskákat kódolt, amilyen például a névjegykártya-nyilvántartó. Láthatjuk, többre is képesek, csak legyen, aki megfizeti őket.

Olvasom a Byte-ban, hogy a GUI-t felváltja a NUI, azaz a Graphical User Interface-t a Network User Interface. Tehát már nem a külalak, hanem a tartalom a döntő a kezelői felület megnevezésében. Érdekes pálfordulás a nomenklatúrában. Arccal az Internet felé. A WUI, azaz Web User Interface elnevezést egyszerűen átugrotta a szaksajtó. Miként most én is.

Em(a)il és a detektívek

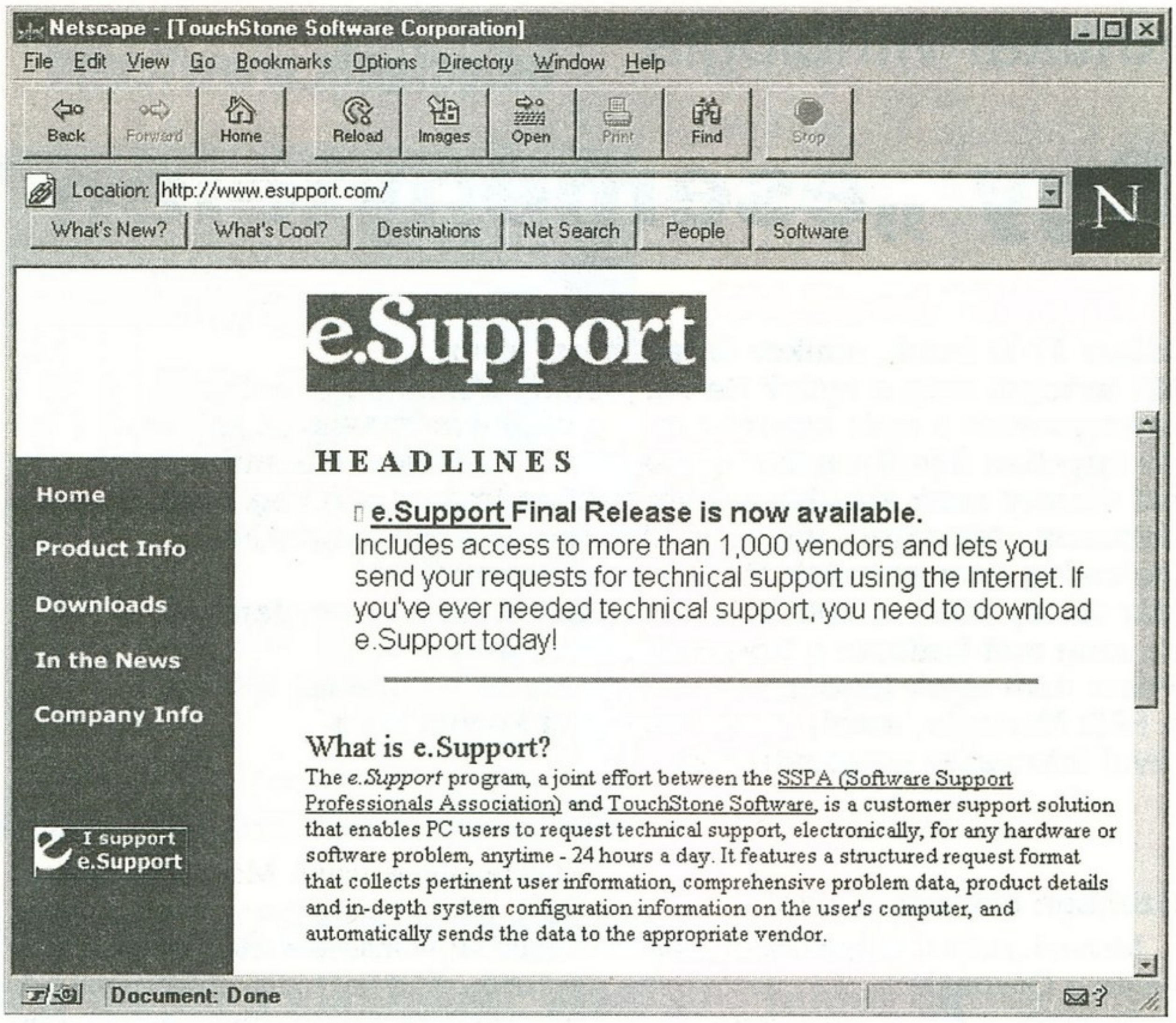
A Checkit nevű diagnosztikai szoftveréről híres és ismert minicég, a kaliforniai Huntington Beach városkájában székelő TouchStone Software gondolt egy nagyot (de tényleg nagyot), és kiagyalta, hogy a Neten keresztül online, azonnali segítséget nyújt azoknak, akik kérik. El is nevezte szolgáltatását e.Supportnak. Egyúttal e-mailben elküldhetjük a gépünkről készített diagnosztikai fájlt is, ami egyértelműen megmondja egy avatott guru számára, mi is van a PC-nkben mind hardver-, mind pedig szoftverszempontból. Az adatbázis több mint 1000 gyártót sorakoztat fel, ezek e-mail útján azonnal elérhetők. De hát ez ugyebár még nem online. Aki többet is meg akar tudni a dologról, látogasson el a www.esupport.com-ra. Nem kell megijedni, tényleg ingyenes.

Meggyőző (de a felhasználó iránti kötelező tisztelet és kímélet miatt elég kis méretű) animációk készítése a Webre nem is olyan egyszerű dolog. Ezernyi shareware ingyombingyom található úton-útfélen, de egyik sem az igazi. Az amerikai Macromedia cég (többek között a Freehand és Director gyártója) bevásárolta, és új fantázianevén, Flash 2 fedőnéven árulja a GIF-es animációkészítés magasiskolájának számító, rendkívül produktív, mintegy 200 dolláros termékét.

Aranyos kis animációkat mindenki szeretne készíteni, de ne felejtsük el, hogy ennek már vannak nem mindenki által megtanulható elemei is. Hogy ne kisebbet említsek, maga Walt Disney is ezen a pályán focizott valaha. Jó adag fantázia kell hozzá, meg ízlés, meg valódi kreativitás…

Life in the Universe

Stephen Hawking, akit az élet már fiatalon tolószékbe kényszerített, intuitív, közérthető módon mesél az élet keletkezéséről, annak fizikájáról és matematikájáról, a csillagokról, a Marsról stb… Nem fogják elhinni, melyik cég adta ki ezt a mindenkinek nyugodtan ajánlható ismeretterjesztő, tudomány-népszerűsítő CD-t: a MetaTools, amely eddig csak a profiknak és álamatőröknek készült kreatív grafikai programok terén vitézkedett (Kai's PowerTools, Kai's PowerGoo). Most azonban mindenki számára izgalmasat alkotott. Klassz lenne, ha valaki magyar nyelvre is lefordítaná, bár angolul is kiválóan élvezhető.



A Web Teleporter segítségével további információkat habzsolhatunk magunkba. Evés közben jön meg az étvágy. Amúgy a prezentáció a Macromedia cég Directorának segítségével készült, annak Lingo nevű scriptnyelvén. Gondolom, már senki sem lepődik meg azon, hogy a CD-n ott található az Apple cég Quick Time 2.1.2 változata mint megjelenítő, továbbá a Microsoft Internet Explorer 3.02 azoknak, akiknek még nem lenne meg. Doppler effektus, élet(?) a Marson, pulzárok, ősrobbanás, kvazárok, fekete lyukak, az élet valószínűsége Földünkön kívül, a világegyetem evolúciója és kronológiája, hogy csak néhány érdekesebb címszót említsek.

TextPad 3.0.08

Végül, de nem utolsósorban bemutatok egy folyamatosan fejlődő, csinosodó, okosodó, európai, még pontosabban angliai eredetű windowsos texteditort, egy valódi 32 bites programot. Legígéretesebb szolgáltatásai a következők:

— Többablakos, ugyanannak az állománynak különböző részeit egyszerre is megmutató program.

— A unixos világból megszokott reguláris kifejezések szerinti, azaz spéci dzsókeres keresés és csere.

— Oszlopszerű blokkokat is képes kezelni.

— Helyesírás-ellenőrzés.

— DOS-os ASCII és windowsos OEM ANSI karaktertábla kezelése, közöttük egylépéses konverzió.

— OLE2 drag and drop.

— Sorba rendezés.

— Billentyűmakrózhatóság.

— Beépített HTML-kezelés.

— ASCII, HTML-tábla segédlet.

— Kérésre sorszámozott sorok, nem látható vezérlőkarakterek kijelzése.

— Az editor megjelenésében is szabadon átszabható (megrögzött BRIEF vagy WordStar felhasználók számára ezek emulációja).

— Egyszerre több fájlon keresztüli keresés.

— Állományok komparálása, szinkronizációja.

— Könyvjelzők, betűk vagy szavak felcserélése (transzponálása).

Szubjektív vélemény: helyre kis elegáns és kényelmes editor. És szemmel láthatóan a fejlesztők még nem végeztek…

Herczeg József

Rövid hírek az OS/2 világából

RTF-ből IPF-be

A jelenleg bétatesztelés alatt álló Rtf2Ipf segédprogram bizonyára nagyon hasznos lesz azok számára, akik gyakran készítenek INF (információs) fájlokat. Az Rtf2Ipf ugyanis automatikusan alakítja át az RTF (Microsoft WinHelp Rich Text Format) formátumban készített anyagokat IPF nyelvre, amelyből aztán egy megfelelő fordítóval (IPFC) készül az INF fájl. A tesztverzió a http://www.backswc.com/rtf2ipf.htm oldalon található.

VisualAge for Java 1.0

Az IBM kiadta ezt az első verziót, melynek ún. Entry változata ingyenes az OS/2 felhasználók számára, és a http://www.software.ibm.com/ad/ vajava oldalról tölthető le. A továbbfejlesztett (Professional) változatot 84,95 dollárért kínálja a Java SuperStore (http://www .javasuperstore.com), s létezik nagyvállalatok számára készített kiadás (Enterprise Edition) is. A vizuális fejlesztőeszköz hardverigénye meglehetősen magas, mivel Pentium processzorra és 32–64 Mbájt RAM-ra van szükség a megfelelő működéshez. A fejlesztőeszközzel egyidejűleg bocsátotta ki a Taligent cég (http://www.taligent.com) az IBM megbízásából készített VisualAge WebRunnert, ennek segítségével igen rövid idő alatt lehet Web-applikációkat fejleszteni. Tovább könnyítheti a Java-programozók munkáját az Oberon Software (http://www .OberonSoftware.com) által piacra dobott Java Toolkit is.

BBBS v3.33

A BBS-t futtatók és használók körében valószínűleg jól ismert a BBBS program, amelynek most jelent meg a 3.33-as kiadása. Az OS/2 mellett a program nagyon sok más operációs rendszeren (például Windows NT-n, Linuxon, DOS-on) is fut. A BBBS azoknak is ajánlott, akik még csak most tervezik, hogy BBS-t alapítanak, mivel a csomagban gyakorlatilag mindent megtalálnak (például teljes Internet- és FidoNet-támogatást), amire szükségük lehet. A BBBS OS/2-es változata az ftp://ftp.bbbs.net/pub/dist/bbbs/bbbs_2 .zip fájlban lelhető fel.

Newsbeat 1.00

A PMINews nem sokáig élvezhette a legújabb OS/2-es hírcsoportolvasó program címet, mivel néhány nappal kibocsátása után megjelent a már igen régen beharangozott Newsbeat is. A binary hírcsoportokat látogatók minden bizonnyal nagyon fognak örülni a Newsbeatnek, mivel integrálható a PMView-val, és lehetővé teszi a kódolt állományok (uuencode) előzetes megtekintését is. A Newsbeat honlapja: http://www.cyberbeach.net/~minogue /newsbeat.html.

Grafikus programok jönnek és mennek

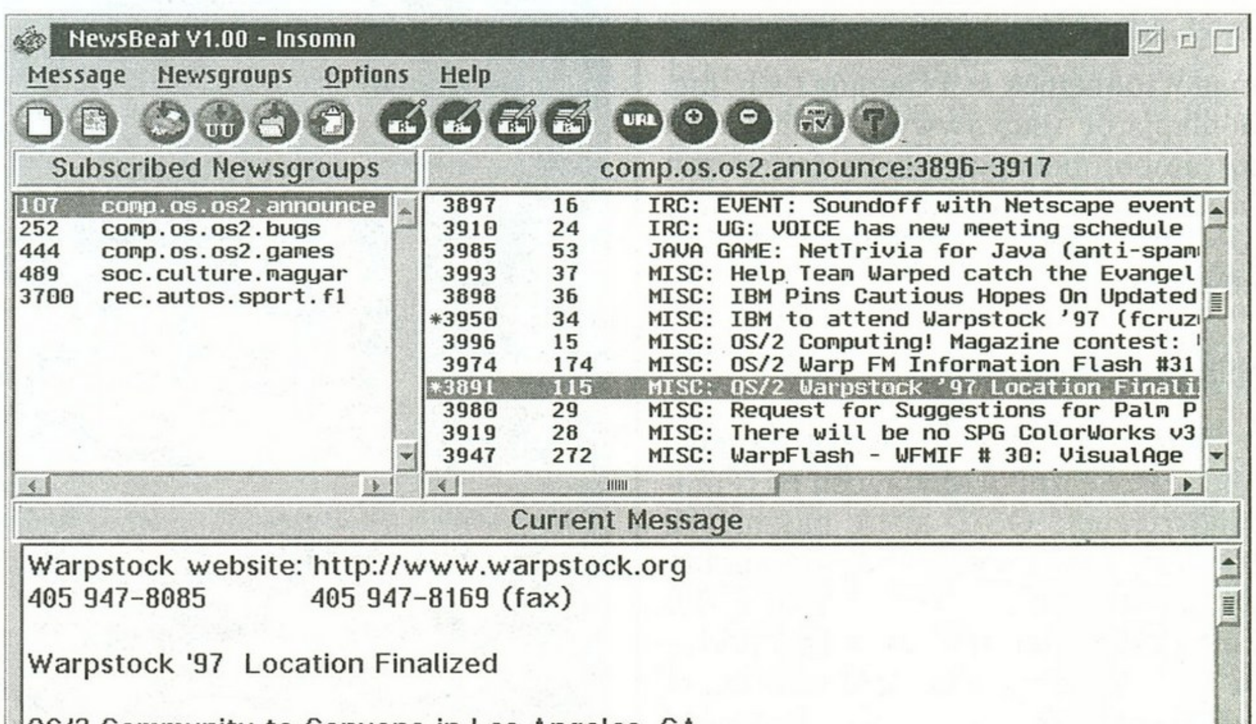
Az előzetes híresztelésekkel ellentétben sajnos nem lesz OS/2-es változata a ColorWorks v3-nak. Túlságosan szomorkodnunk azonban nem kell, mivel a már említett Photo>Graphics PRO v2 mellett egy újabb grafikus program, az MD+F SX Paint is megjelent a piacon. A Modular Dreams Incorporated új termékét ingyen adják a néhány héttel ezelőtt bejelentett MD+F WebAK programmal, amelyet Web-grafikák készítésére fejlesztettek ki. Mindkét programról képeket és további információt találhatunk a Modular Dreams honlapján: http://www.modulardreams.com. További jó hír, hogy a ColorWorks for OS/2 tulajdonosainak árengedményt ad a Dadaware, amennyiben áttérnek a cég Embellish termékére. Az Embellish 2.02b bétaverziója a közelmúltban jelent meg (ftp://ftp.dadaware.com/dadaware/embobeta.zip).

A Navigator javítása

A Netscape böngészőiben június közepén biztonsági hibát fedeztek fel, amely lehetővé tette, hogy a Web-kiszolgáló (JavaScript vagy Form oldalak böngészése közben) lokális fájlokat töltsön le a kliensgépről. A Netscape Corporation gyorsan elkészítette a javítást, amelyet a böngésző OS/2-es változatába is beépítettek. A javított böngésző (amely ezután azt is lehetővé teszi, hogy a programból négynél több példány is futhasson egyszerre) a szokásos helyről, a http://www.internet .ibm.com/browsers/netscape/warp oldalról tölthető le.

További játékáradat

Az OS/2-es játékok kínálata tovább bővült a Stardock PlusPak részeként megjelent BUGS-szal, amely a hagyományos „lövöldözős" játékok sorát (például Galaga) folytatja. A játék lényege az, hogy meg kell védenünk a Földet a világuralomra törekvő Bill ördög szörnyűséges űrhajóinak (Bill's



Unbelievable Ghastly Spaceships, innen jön a BUGS) garázdálkodásaitól. A CodeSmith Software fejlesztésének eredményeként jelent meg a Tetrishez hasonló, Matrix nevű stratégiai játék. A BUGS-szal ellentétben a Matrixnak shareware és teljesen funkcionális próbaváltozata tölthető le a http://www.bmtmicro.com/catalog/matrix/matrix.html oldalról.

LXOPT v1.22

Az OS/2 alá fejlesztők valószínűleg nagyon fognak örülni annak, hogy Mike Ruddy kiadta 32 bites kódoptimalizáló programjának teljesen ingyenes (freeware) változatát. Megtalálható az ftp://hobbes.nmsu.edu/pub/os2/dev/util/lxopt122.zip fájlban.

Chorus/2

Hét nagyon hasznos, többségében már régóta ismert és sokak által használt kiegészítő programot integrált egy csomagba a BMT Micro, Inc. A Chorus/2 kiadásának az volt a célja, hogy a felhasználók feltétlenül felfigyeljenek a „becsomagolt" alkalmazásokra: X-IT (Munkaasztal-kiegészítő), DragText (a Drag&Drop technológia kiterjesztője), Smalled (gyors WYSIWYG szövegszerkesztő), SMEHTM Extensions (kiterjesztés a Smalledhez, amellyel az HTML szerkesztővé alakítható), SwapMonitor (tárcserefájl-figyelő), FileFreedom (PM fájlmenedzser), valamint az előző hírben már említett Matrix játék. További információ: http://www.bmtmicro.com/chorus.

GammaTech IRC Fixpak

A GtIrc 2.x-tulajdonosok javítócsomagot tölthetnek le a GammaTech, Inc. honlapjáról (http://www.gt-online.com) az utóbbi hónapok során felfedezett problémák orvoslására. A frissítés után az IRC kliens verziószáma 2.06-ra emelkedik.

HFS segédprogramok

Marcus Better (Marcus.Better@abc.se) elkészült a Macintosh HFS fájlrendszerének OS/2 alatti használatát lehetővé tevő ingyenes segédprogramgyűjteményével. A HFSUTILS a Robert Leslie által írt Unix verzió portja. A http://www.student.nada.kth.se/~f96-bet/ hfsutils oldalról letölthető fájlban Win32-es változat is található az OS/2-es mellett.

Kádár Zsolt

DOS Doktor V.

Ha vészhelyzet van

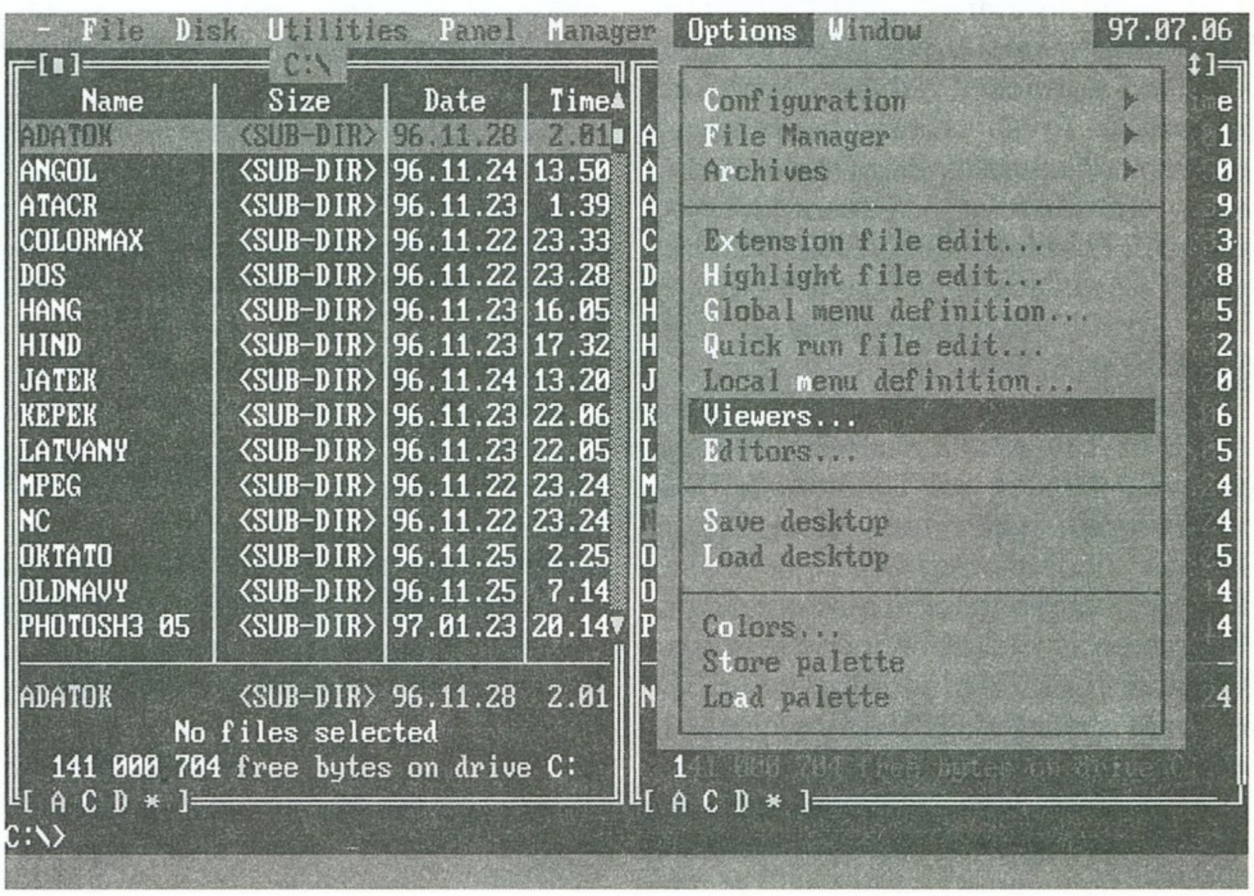
Idei olvasói felmérésünkből kiderült, hogy általános tetszést aratott a tavasszal elindított DOS Doktor-sorozat, és hogy sokan olvasnának ilyesmit gyakrabban is. Most tehát folytatjuk a sorozatot, melynek legutóbbi része júliusi számunkban jelent meg. A tippek és trükkök egyaránt szólhatnak kezdőkhöz és profikhoz, a hasznos hétköznapi tapasztalat (vagy annak felfrissítése) mindenkinek jól jöhet.

Korábban már foglalkoztam a minden bizonnyal legnépszerűbb két fájlkezelővel, és megpróbáltam röviden összehasonlítani legjellemzőbb tulajdonságaikat. Ezúttal ismét egy ilyen keretprogramról lesz szó, a DOS Navigatorról, annak is egy bizonyos funkciójáról, a viewerről (számomra kedves magyar fordítása a „nézőke"). Akkoriban azt mondtam, hogy a Norton Commander nézőkéje intelligensebb. Most viszont éppen azt szeretném bemutatni, hogyan tehetjük a DN nézőkéjét rugalmasabbá, intelligensebbé.

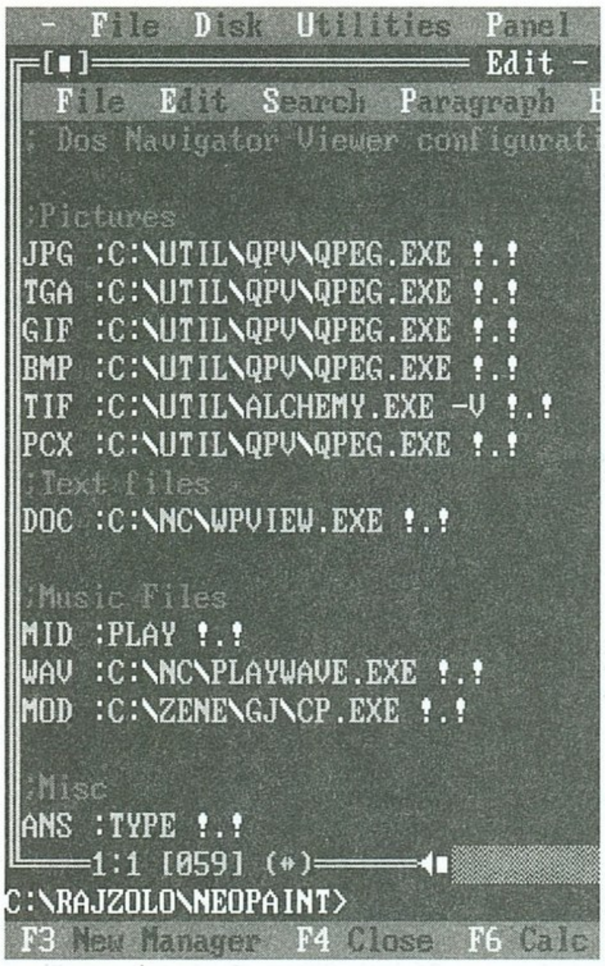
Menü és paraméterezés

A fájlkezelő keretprogramok többnyire az F3 funkcióbillentyűvel teszik lehetővé, hogy egy adott fájlba betekintsünk, és megnézzük annak tartalmát. Formátum nélküli szövegfájlok (például .TXT, .BAT kiterjesztésűek) esetében ez általában elég egyszerű, hiszen azok jól áttekinthetőek. Programfájlok, programok által használt adat- és segédfájlok esetében már csak karakterek, szimbólumok, írásjelek halmazát látjuk, illetve átkapcsolhatunk hexadecimális megjelenítésre, de mindez csak a programozókat érdekli. Az átlagfelhasználó számára leginkább az a fontos, hogyan tud elolvasni egy bármilyen szövegszerkesztővel előállított szövegfájlt, megnézni egy képfájlt, vagy meghallgatni egy hangfájlt.

A DN nézőkéje konfigurálható, megadhatjuk benne, hogy adott kiterjesztésű fájlokat a rendelkezésünkre álló segédprogramok melyikével jelenítse meg. A DN fejlécén kattintsunk rá az Options menüre, vagy használjuk az ALT+O billentyűkombinációt. Ekkor megjelenik az Options legördülő menüje. (1. kép.)



1. kép



2. kép

Most kattintsunk rá a viewers menüpontra, vagy nyomjuk le a „v" billentyűt, ekkor megjelenik a nézegető szerkesztője. (2. kép.)

Most már csak annyi dolgunk van, hogy a megfelelő helyre beírjuk a képen látható minta szerint az elérési úttal együtt a kívánt program nevét. Amennyiben paraméterezésre is szükség van, ne felejtsük el megadni! Ha rosszul paramétereztünk, hibásan jelenhet meg például egy kép, vagy csak a programkódot fogjuk látni, esetleg nem történik semmi, visszatérünk a fájlkezelőbe.

Mint látható, minden kiterjesztésre megadhatjuk a megfelelő segédprogramot, és a DN az F3 billentyű lenyomására ezt fogja használni. A fenti példa esetében például a .DOC kiterjesztésű fájlok olvasására a DN meghívja a lemezünkön található Norton Commander nézőkéjét, s így az ott található gomb segítségével kiválaszthatjuk a megfelelő szerkesztő megjelenítését. Ha valamelyik hangfájlra vagyunk kíváncsiak, akkor a DN a meghívott segédprogrammal lejátssza azt, ha egy képfájlba akarunk bekukkantani, akkor a kiterjesztés szerint megadott segédprogrammal megjeleníti a képet. Nem kell többé bemászni a segédprogramot tartalmazó könyvtárba, ott elindítani a programot, majd külön kikeresni a megjeleníteni kívánt fájlt, mindez automatikusan végbemegy az F3 billentyű lenyomására!

Ha több fájlt akarunk egymás után megnézni vagy meghallgatni, akkor az ESC gombbal vissza kell térnünk a DN-be, és a kurzorral rá kell állnunk a fájlra, azután újból az F3-at kell lenyomnunk.

A fentiek segítségével tehát bárki maga konfigurálhatja a DN-jét a fájlok megjelenítésére, illetve meghallgatására, a rendelkezésére álló segédprogramok használatával. Az én DN-em a mellékelt képen látható konfigurálással közel húszféle szövegszerkesztő által előállított szövegfájlt tud olvashatóvá konvertálni, továbbá képes megjeleníteni a JPG, PCX, BMP, TIF, GIF, TGA kiterjesztésű képállományokat, illetve lejátssza a WAV és a MID kiterjesztésű zenei és hangfájlokat. A képfájlok megjelenítésére a QPEG, illetve a QPV shareware programokat ajánlom, a hangfájlok lejátszására a SoundBlaster hangkártyák play.exe állománya kiválóan alkalmas.

Bizonyos szövegszerkesztők által előállított fájlokba külső nézőkével betekintve némileg összezagyválva látjuk a bekezdéseket, a szöveget, de ez nem hiba (legfeljebb udvariatlanság), mert maga a szövegszerkesztő a mondatokat pontosan a helyükre rakva mutatja és nyomtatja, csupán a fájlon belül ilyen slendrián módon tárolja. A fájlbetekintés ilyen esetekben tehát csak hozzávetőleges tájékozódásra alkalmas, pontos szövegolvasásra nem nagyon. (A létrehozó program egy kis trükkel nélkülözhetetlenné tette magát.)

Ha gondok vannak

Egészen más terület. Sajnos, néha előfordul, hogy DOS alatt induló számítógépünk nem óhajt normálisan startolni. Korábban foglalkoztam már ezzel, és tanácsokat is adtam erre a problémára. Most egy egészen egyszerű és kényelmes megoldást ajánlok floppylemezről történő indításra. Elsősorban azok számára ajánlom, akik rendelkeznek a DOS Navigator nevezetű fájlkezelő keretprogrammal. (Akinek nincs meg, annak feltétlenül ajánlom mielőbbi beszerzését, majd látni fogjuk, miért...) Most egy olyan, univerzális indítólemez készítését írom le, amellyel bármelyik (!) DOS alatt induló számítógépet el lehet indítani, tehát nem csak a saját gépünket.

Először is, vegyünk elő egy hibátlan, jó minőségű floppyt (amelyet le is ellenőriztünk), majd a format a: /s paranccsal készítsünk belőle rendszerlemezt. Ha ez megtörtént, akkor másoljuk rá a lemezmellékleten található autoexec.bat és config.sys állományokat, valamint a mouse.com állományt, amely egyébként a DOS könyvtárban mindenkinél rendelkezésre áll. Ez utóbbi azért kell, hogy az egerünk is működni tudjon lemezről. Ha mindezzel végeztünk, másoljuk át az egész DN könyvtárat. (Az NC-t, azaz a Norton Commandert nem is tudnánk, hiszen ez nem fér rá még egy üres lemezre sem!) Tessék, ki lehet próbálni, a lemezről úgy indul el a gép, hogy nem kell vesződni a DOS parancsok beírásával, hiszen mindjárt a DN fájlkezelőben mozoghatunk, s az egeret is használhatjuk ehhez.

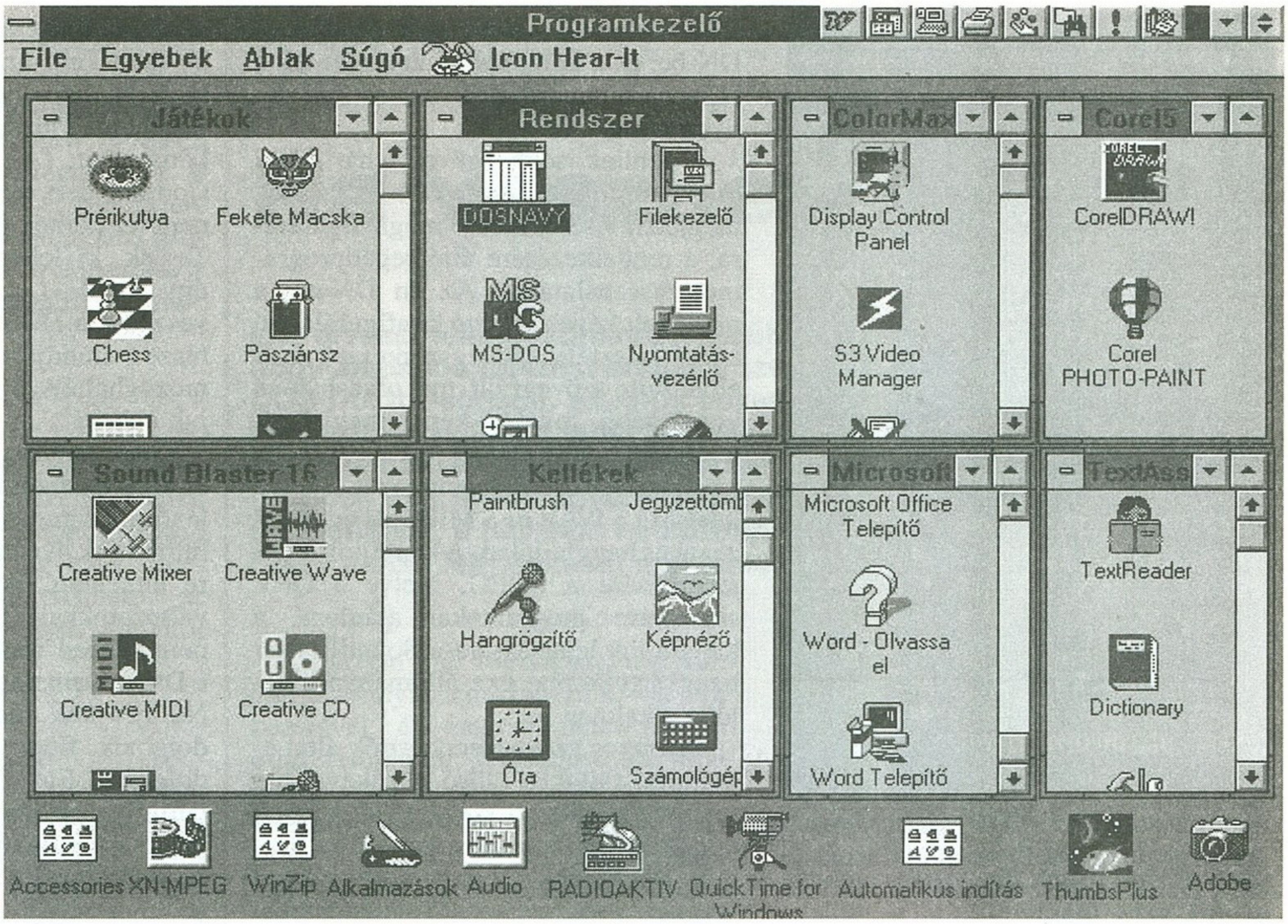
Innen kezdve már megjeleníthetjük és szerkeszthetjük a merevlemezünkön lévő konfigurációs fájlokat, illetve felülírhatjuk őket az előzetesen dugihelyre elmentett, korábban még működő változatukkal! (Figyelem! Ezt a lemezt nem szabad írásvédetté tennünk, mert a DN folyamatosan ír az indítólemezre! Ne ijedjünk meg, ha nagyon lassan dolgozik floppylemezről, mert azért dolgozik! Egy ilyen lemez birtokában bármilyen szoftverhiba esetében el tudjuk majd indítani a gépünket — vagy ha szükséges, a másét.)

Segít az NU

Akkor már nagyobb baj van, ha az univerziális indítólemezzel sem indul el a gép... Bizony ilyen is előfordul, főként ha lemerült a BIOS-adatokat tápláló elem, vagy tönkrement az akkumulátor — régebbi gépeknél elem táplálja, az újabbaknál akkumulátor, amely a hálózatról újratöltődik. Ez a probléma könnyen felismerhető, a korai szakaszban mindössze annyi történik, hogy gépünk „elfelejti" a rendszeridőt, s ezt folyton utána kell állítani. Ha viszont végleg lemerült az elem, illetve tönkrement az akkumulátor, akkor gépünk nem ismeri fel még a merevlemezt sem. Néha még az is előfordul, hogy bizonyos programok „belenyúlnak" a BIOS SETUP-ba, és átírják az itt tárolt adatokat. Az átlagfelhasználó persze nem mer belepiszkálni a BIOS SETUP-ba — minek is, ha nem ért hozzá —, de még a tájékozottabbaknak is kell egy kis idő, mire helyre tudják állítani az adatokat. (És nem is mindig sikerül elsőre, hiszen elég elütni egy billentyűt...) Ha az elem/akkumulátor volt a hibás, akkor azt ki kell cserélni, illetve cseréltetni. Utána pedig helyreállítani a BIOS SETUP adatait. A szervizes szakemberek is hálásak, ha ettől megkíméljük őket, hiszen magunk is megtehetjük, ha rendelkezünk a megfelelő lemezzel...

A Symantec Norton Utilities 8.0 programcsomagja (NU8) segítségével elkészíthetjük ezt a lemezt. Lépjünk be

az NU8 könyvtárba és gépeljük be: norton. Ekkor megjelenik a főmenü, amelyből válasszuk a rescue disket. Ez a kiválasztott floppymeghajtóban lévő lemezre elkészíti a visszaállító programot. Ebben rögzítődik gépünk hardver- és szoftverkonfigurációja. Tulajdonképpen egy speciális indítólemezt készít, amelyet akkor kell felhasználnunk, amikor az előbb említett problémák miatt nem tudjuk használni gépünket. Ilyen esetben elég behelyezni ezt a lemezt és erről indítani. A megjelenő promptnál be kell írni: Rescue, s a menüből a Restore-t kell választani. Ekkor újabb menü jelenik meg, s itt be kell ikszelnünk, hogy milyen adatokat kívánunk visszaállítani. Kezdőknek ajánlom mindegyik négyzet ikszelését. Ekkor ez a nagyszerű program visszaállítja a korábban elmentett hardver- és szoftverkonfigurációt, és ezután már használhatjuk is gépünket.



3. kép

Figyelem! Minden hardverváltozáskor (új hangkártya telepítése, új videokártya, más monitor stb.), valamint az autoexec.bat és a config.sys állományokat érintő változások esetén újra el kell készíteni ezt a lemezt! Ha nem mulasztjuk el, akkora a rescue (mentő) lemez minden olyan esetben helyre tudja állítani gépünk normális állapotát, amikor nincs hardverhiba. Ezt a lemezt viszont a DN-lemezzel ellentétben tegyük írásvédetté.

DOS Navigator — Windows alatt

Nemigen találkoztam még Windows alatt igazán jó fájlkezelővel — legalábbis olyannal nem, amelyik minden tekintetben felvenné a versenyt a Norton Commanderrel vagy pláne a DOS Navigatorral. (Az NC-nek viszont van windowsos változata.) Próbálkoztam már a Wincommander-rel, de nem nyerte el igazán a tetszésemet.

Nos, én elhelyeztem egy ikont a Windows For Workgroups 3.11 Rendszer nevű csoportablakában (persze bárhová tehettem volna), és innen futtatom a DOS Navigatort. Az 1.33-as és az 1.42-es verzió (az utóbbi 1996-ban készült) egyaránt kifogástalanul működik!

Minden funkciója használható, még a Shift+1 és Shift+2 gombokkal történő tömörítés-kicsomagolás is. Gyorsan és flottul mindenféle művelet elvégezhető vele. Korábban a Windows 3.1 verzióval is kipróbáltam, ott azonban törlésnél a Windows letiltott, érvénytelen műveletre hivatkozva, és a tömörítés-kicsomagolás sem működött. A Win 95-nél pedig ki kell kapcsolni a DOS Navigator Options legördülő menüjében a Configuration almenüben a System Setupban az elérni kívánt meghajtókra a közvetlen hozzáférést (Direct access), és ugyanitt be kell kapcsolni az OS-függő lemezkezelést (OS-dependent Disk Access).

Ugyanezt érdemes kipróbálni a Win 3.1 verziónál is — én ezt akkor nem tettem meg. Aki úgy dönt, hogy Windows alatt sem válik meg kedvenc DN-jétől, annak kellemes navigálást!

Mózes István Miklós

OKTÓBERI SZÁMUNKBAN A HÓNAP TÉMÁJA:

JOGINFORMATIKA

Hűség és technikai haladás

Irány a CD-ROM!

Idei olvasói felmérésünk egyik kérdése alapján arra vártunk választ olvasóinktól, hogy mennyire helyeslik vagy ellenzik az Új Alaplap állandó mellékletének CD-re változtatását. A közvéleménykutatás 1579 kitöltött kérdőívéből ezt a „részletkérdést" vettük elsőként nagyító alá, hogy mielőbb döntéselőkészítő információk birtokába jussunk.

Lapunk hovatovább 15 éves történetében volt néhány lényeges átalakulás. Utólag bebizonyosodott, hogy ezek a lépések nemcsak a (mindenkori) kiadó és a (szinte változatlan összetételű) szerkesztőség szándéka szerint, hanem tényleges hatásukat tekintve is a folyamatosságot és a fejlődést szolgálták.

A változtatások kiváltó oka többnyire valamilyen új technikai lehetőség megjelenése volt, és a számítástechnikai lapok közül majdnem minden fontos újítás bevezetésében mi voltunk az elsők — egészen a legutóbbi időkig, amikor is az állandó CD-ROM mellékletre való átállásnak és a saját weblap létesítésének „frontján" eleve lemondtunk a zászlóvivői szerepről.

A CD-ROM-mal kapcsolatos álláspontunknak többször hangot is adtunk. Megírtuk, hogy az Új Alaplap CD-jének méltónak kell lennie a nyomtatott lap igényes tartalmához: csak akkor szabad előrukkolni vele, ha a minőségi olvasói igények kielégítése biztosítottnak látszik. A mi CD-nk ne legyen hevenyészett programraktár, és ne legyen tele semmitmondó helytöltögetőkkel sem. Nekünk a CD-n is értelmes, hasznos, jó és korszerű dolgokat illik olvasóinknak átnyújtani.

Ez a feladat ugyanakkor messze meghaladja belső szerkesztőségi kapacitásainkat, ami szükségessé tette egy velünk együttgondolkodni kész, hozzáértő külső gárda felkutatását és bevonását. Ezt a profi csapatot most megtalálni véljük az OS/2 Times c. lapot kiadó OpenBlue Bt tagjaiban, akik a CD tartalmi és technikai gondozását teljeskörűen vállalni tudják. (Vezetőjüknek, Ambrózy Gábornak a neve bizonyára olvasóink előtt is ismerősen cseng.)

Bizonytalanságunk továbbá azzal függött össze, hogy valahányszor egy lapnál sor kerül lényeges tartalmi vagy formai változtatásra, annak elkerülhetetlenül vannak vesztesei is. Esetünkben azok kerülhetnek ilyen helyzetbe, akik körülményeik miatt nem tudnak lépést tartani a számítástechnikai eszközök megújításának követelményével.

A számítástechnikai lapok közül minden jel szerint az Új Alaplapnak van a leghűségesebb olvasótábora. Ez bennünket is az olvasótábor iránti hűségre kötelez. Arra, hogy a többség akaratának érvényesítése mellett a lehetőségek határai között figyeljünk a kisebbségre is. Nem állítjuk, hogy ez az új kihívás, a lemezmelléklet „CD-sítése" ugyanolyan nagy horderejű lépés, mint például az egykori Mikroszámítógép Magazin Alaplappá alakítása volt, de a párhuzam kézenfekvő: úgy kell a változtatást megoldani, hogy minél több olvasónkat megtartsuk, minél kevesebben érezzék magukat hirtelen a leszakadók, a „cserbenhagyottak" között.

Az 1997. áprilisi számunkban közreadott kérdőíven ezért is kérdeztük meg olvasóinkat, hogy egyrészt hogyan vélekednek a CD-ROM melléklet rendszeresítéséről, másrészt milyen arányban van CD-meghajtójuk. A válaszok bizony feladták nekünk a leckét! Az igennel szavazók aránya 60 százalék volt, tehát a határozott többség a CD mellett voksolt. Elutasította az állandó CD-t az olvasók 15 százaléka, míg a kisebbség nagyobbik fele (25 százalék) azt a kombinált megoldást javasolta, hogy a CD-re való átállás mellett maradjon egy előfizethető floppys változat is. A felmérésben azért nem adtunk meg ennél többféle válaszlehetőséget, mert hiába lennének esetleg másmilyen igények is, havi rendszerességgel finanszírozási, szervezési, technikai stb. okokból nem tudnánk vállalni olyan megoldást, hogy például egyszerre legyen a lapban CD is és floppy is (ahogy eddig volt alkalmanként), vagy hogy az utcai árusításban is kétféle (CD-s és floppys) változat legyen kapható.

Az állandó CD-t igenlő szavazatokat megerősíti a felmérésnek a gépfelszereltségre vonatkozó összesített adata is, mely szerint olvasóink 56 százalékának már otthoni gépében is van CD-meghajtó. Miután pedig a korszerűbb technikára való áttérés az új gépek beszerzésével és a meglévők feljavításával Magyarországon is tartós folyamat, a CD-ROM-on mellékelt anyaghoz való kényelmes hozzáférés mind a munkahelyeken és az iskolákban, mind az otthoni gépeken egyre inkább magától értetődő lesz.

Következő (1997. októberi) számunkban még az eddigi gyakorlatnak megfelelően, tehát a szokásos floppy mellett extraként lesz ott a CD melléklet, de ez már tartalmilag és formailag a rendszeres CD-ROM előhírnöke, próbaszáma, „nulladik" változata... A floppy helyére 1998 januárjától pedig minden számban a korong lép.

A kisebbséggel pedig ugyancsak törődve olvasótáborunknak a CD-t fogadni egyelőre még nem tudó hányada részére az Új Alaplapból 1998 januárjától külön előfizethető floppy mellékletes változat is lesz, és ezt a „dualizmust" mindaddig fenntartjuk, amíg a floppys előfizetésekből annak különköltségei legalább nullszaldósan megtérülnek.

Faklen Pál

Meglepetés-CD szeptemberben

Bizonyára meglepődtek azok, akik a szeptemberi Új Alaplapban mindenféle beharangozás, reklám és felár nélkül találtak egy „extra" CD-ROM mellékletet. A meglepetésért a székesfehérvári Axis cég a „felelős". Néhány hónapja állították össze azt a CD-t, amely az általuk képviselt Sybase Inc. szoftvereit hivatott bemutatni, és amelyből mintegy 3000 példányt módjukban állt lapunk olvasóihoz is eljuttatni.

Néhány csemege a CD tartalmából: PowerBuilder — animáció; S-Designor 5.0 CASE-eszközcsalád — korlátozott, működő szoftver; Sybase SQL Anywhere — SQL szerver; JATO-RAD Java-fejlesztőeszköz — bétaváltozat; Optima++ 1.5-RAD — C-fejlesztőeszköz, futtatható változat. A néhány hónapnyi késés miatt egy-két „lejárt" program is van a CD-n, de a jól használható, kipróbálható eszközök mellett ez inkább csak kisebb szépséghiba.

A 3000 CD-ből természetesen nem jutott lapunk minden vásárlójának, és magától értetődően az előfizetők voltak a kedvezményezettek, de az Új Alaplapot a hírlapárusoknál megvásárlóknak is van lehetőségük a CD beszerzésére: az Axis a náluk jelentkezőknek is elküldi azt — korlátozott számban, amíg a készlet tart. (Telefon: 06/22/327-631)

Rekordnyereség a Sunnál

A hálózati technológiák konjunktúráját jól tükrözi, hogy rekordnegyedévről, illetve rekordévről számolt be a Sun Microsystems. A június végén zárult üzleti év 8,6 milliárd dollárnyi árbevétellel zárult, ami 21 százalékos növekedés 1996-hoz képest, ezen belül a negyedik negyedév volt a legerősebb a maga 2,5 milliárd dolláros rekordjával. Ugyancsak a negyedik negyedévhez fűződik a másik rekord: a részvényenkénti nyereség elérte a 61 centet (szemben például az előző évi 31 dollárcenttel). Az egész év nettó nyeresége egyébként 762 millió dollár volt.

Újdonságok a Microsofttól

A Microsoft kibocsátotta az Internet Explorer (IE) 4.0 verziójának második bétaváltozatát. Az új változat a felhasználóknak teljes kommunikációs és együttműködési megoldást, valamint webintegrációt ajánl. Megjelent a Proxy Server 2.0 bétaváltozata is, ez az első bővíthető tűzfal- és integrált webgyorsítótárszerver a Windows NT Server 4.0 változatához. (Ingyenesen letölthető a http://www.microsoft.com/proxy/beta címről.) Július közepén jelent meg a Windows NT Server, Enterprise Edition 4.0 bétaváltozata, amelyet világszerte több mint 2100 felhasználónak és gyártónak küldenek el, s benne lesz a Microsoft Developer Network (MSDN) Level 3. változatának következő kiadásában is. A bétaváltozat tartalmazza a Microsoft Cluster Server (előzőleg: „Wolfpack"), a Microsoft Message Queue Server („Falcon") és a Microsoft Transaction Server („Viper") frissítéseit is.

Borland: mínuszból plusz

Az elmúlt évek veszteségeihez képest a Borland végre szerény nyereséggel zárta 1998-as üzleti évének 1997. június 30-án zárult első negyedét (a sajátságos gazdálkodási időszámítás miatt 1997-es pénzügyi évük március 31-ig tart). Igaz ugyan, hogy nettó profitjuk csak 79 ezer dollár volt, de ez már profit, és jelzi a cégen belüli radikális változtatások eredményét, a stabilitás elérését. Ha minden így folytatódik, akkor van rá esély, hogy a Borland a szakma által is indokoltnak tartott vezető helyet érjen el a fejlesztőeszközök és a fejlesztési technológiák piacán.

WAP és HDML

Egy júliusi hír értelmében az Ericsson, a Motorola, a Nokia és az Unwired Planet a jövőben fokozott támogatást nyújt a vezeték nélküli alkalmazási protokoll kifejlesztéséhez. (Wireless Application Protocol = WAP.) A kezdeményezést rajtuk kívül támogatja az Alcatel, a Mitsubishi Electric, a Nortel, a Philips és a Siemens. Arról is együttesen nyilatkoztak a csoport tagjai, hogy támogatják a hordozható eszközökre vonatkozó jelölési nyelvet (Handheld Device Markup Language = HDML), amely a WAP mikroböngészőjének alapja. A mobiltelefongyártókat a közös szabvány arra bátoríthatja, hogy befektessenek a fejlett alkalmazásokat támogató, kompatibilis hardvereszközök fejlesztésébe. A protokoll tervezetét szeptember 15-től tervezik nyilvánossá tenni a Weben.

A Synergon jól rajtolt

Az Optotrans és a Rolitron egyesüléséből létrejött Synergon Rt az előzetes terveket felülmúló második negyedévet zárt mintegy 1,2 milliárd forint forgalommal, az első félév adózás előtti nyeresége pedig elérte a 180 millió forintot, ami 200%-os eredményjavulás 1996 hasonló időszakához képest. Az üzleti siker alapján simának tűnik az út a Synergon nyilvános tőzsdei megjelenése előtt is, főként ha figyelembe vesszük azt a tőkebevonást, amelynek keretében a cégnél szeptember végéig egy jelentős pénzügyi csoport is kisebbségi részesedést szerez. A Synergon a maga számára a jövő kulcsfontosságú területeinek a vezeték nélküli kommunikációt, az ATM technológiát, az emelt szintű biztonságtechnikai szolgáltatásokat, az integrált vállalatirányítási rendszereket és a testreszabott szolgáltatásokat tekinti.

Kiállítás Szegeden...

November 13. és 15. között első ízben rendezik meg a Szegedi Nemzetközi Vásár legújabb szakkiállítását, a „2001 — Multimédia" kiállítást. (Alcíme: Az ezredforduló kommunikációja a számítástechnika jövője.) A tervek szerint a szegedi kiállítási terület B pavilonja ad helyet magának a szakkiállításnak, míg a C pavilonban a világon egyedülálló ritkaságokat is felvonultató és Magyarországon eddig teljességében be nem mutatott számítástechnikai old timer kiállítást rendeznek be. Aki a természetes helyett kénytelen beérni a virtuális élményekkel, annak a www.sznv.hu címre kell ellátogatnia.

...és Pécsett is

Éppen, hogy véget ért a I. Pécsi Nemzetközi Kultúrhét keretében lezajlott Drávanet Internet Csarnok című szakkiállítás, amelyen a névadó mellett szponzorként a Matáv és a Bull Pénzügyi és Üzleti Kommunikációs Kft. is közreműködött, máris újabb rendezvény híre érkezett Pécsről. Office'97 néven (na, itt vajon ki lehet a szponzor?) október 9. és 12. között nemzetközi számítástechnikai, irodaszervezési, informatikai és irodaberendezési szakkiállítást és vásárt rendez az Expo-Service a Pécsi Sportcsarnokban. Nyilvánvaló, hogy ez a rendezvény egyelőre nem fenyegeti a röviddel utána nyíló Compfair piaci pozícióit, de ha azt nézzük, hogy Pécsett a kiállítási négyzetméterárak éppen feleannyiba kerülnek, mint a Compfairen, a jövőre nézve a vidéki rendezvények akár valódi konkurenciát jelenthetnek még a legnagyobb fővárosi szakkiállításoknak is.

1200 iskolában Elender

A középiskolai Internet-szolgáltatás kiépítésére kiírt közbeszerzési pályázatot az Elender Kft nyerte el, megelőzve az EuroWeb, a DataNet és a BankNet ajánlatát. A szélessávú (64 Kbites) hálózatra a Művelődési és Közoktatási Minisztérium összesen mintegy 1200 oktatási intézményt fog csatlakoztatni. A hálózat kiépítése ez év őszén kell elkezdeni és 1998 augusztusáig befejezni. A kiszolgálógépeken kívül az iskolákban együttvéve mintegy 9000 számítógépre és egyéb hardverre is szükség lesz, melyekre szintén vonatkozik a nyílt közbeszerzési eljárás. A hardverajánlati dokumentációt 27 cég vette meg, és a pályázat beadási határideje 1997. szeptember 9. Az „internetesedés" kapcsán az új tanévben a diákokon kívül tehát sok tanulnivalóval néznek szembe az oktatók is, nem is szólva a minisztérium előtt álló feladatokról a tartalmi szolgáltatás kidolgozásában.

Főnyeremény a Digitaltól

A közvéleménykutatási kérdőívünk beküldői részére rendezett sorsolás hardver főnyereményének, a Digital Venturis FE számítógépnek a nyertese Szénási Violetta bácsalmási tanítónő (aki jelenleg a városi könyvtárban dolgozik). A nyereményt a Digital Magyarország Kft részéről Kroó Győző kereskedelmi vezérigazgató-helyettes adta át neki.



APRÓHIRDETÉSI MEGRENDELŐLAP

Kérem, hogy az Új Alaplap következő számának
Mikrobazár rovatában közöljék az alábbi szövegű apróhirdetést:

....

....

....

....

....

(Maximális terjedelem: 300 betűhely)

Ingyenes előfizetés az Infopenre

Kérem, hogy regisztráljanak az **Infopen** ingyenes előfizetőjeként az alább megjelölt jogcím(ek)en:

❑ Cégünk rendelkezik kereskedelmi forgalomban kapható Unix rendszerrel.
A Unix verzió: ....

❑ Cégünknél hálózati operációs rendszer működik, több mit 20 felhasználós licenccel.
Az operációs rendszer: ....

❑ Cégünknél a vállalati hálózatnak (LAN) van kapcsolódása az Internetre.
Az adatátviteli sebesség: ....

Cégnév: ....

Postázási cím: ....

....

Név: ....

Telefon: ....

Fax: ....

Dátum: ....

- - - - - - - - - - - - - - - - - - -
/aláírás/

Előfizetés az Új Alaplapra

Az 1997/......... számtól kezdődően előfizetem az Új Alaplap című, havi számítástechnikai folyóiratot ...... példányban, ❑ 1 évre, ❑ 1/2 évre.

Az éves előfizetési díj 4356,– forint.

Az előfizetési díj kiegyenlítéséhez:

❑ **Számlát** kérek (banki átutalással fizetek).

❑ **Átutalási postautalványt** kérek.

Név: ....

(Cég:) ....

Cím: ....

Helység: ....

Irányítószám: ....

Dátum: ....

- - - - - - - - - - - - - - - - - - -
/aláírás/

INFORMÁCIÓKÉRÉS

Az Új Alaplap 1997. szeptemberi számának hirdetéseihez

Kérem, hogy az itt általam **BEKARIKÁZOTT KÓDSZÁMÚ** hirdetésekkel kapcsolatban küldjenek részemre bővebb tájékoztatást.

Beküldhető: 1997. szeptember 30-ig

0901	0917	0933
0902	0918	0934
0903	0919	0935
0904	0920	0936
0905	0921	0937
0906	0922	0938
0907	0923	0939
0908	0924	0940
0909	0925	0941
0910	0926	0942
0911	0927	0943
0912	0928	0944
0913	0929	0945
0914	0930	0946
0915	0931	0947
0916	0932	0948

FELADÓ **Feladáskor kérjük bérmentesíteni!**

A) Egyéni érdeklődő:

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

B) Vállalati érdeklődő:

Cég: ....

Ügyintéző: ....

Cím: ....

Helység: ....

Irányítószám: ....

Telefon/Fax: ....

Bélyeg helye

Új Alaplap

VI., Dózsa György út 84/b

Postafiók 571
Budapest 1539

Bélyeg helye

Új Alaplap

VI., Dózsa György út 84/b

Postafiók 571
Budapest 1539

Minden PC-hez kell egy jó alaplap!

És egy Új Alaplap!

Belföldön díjmentesen feladható

Új Alaplap

VI., Dózsa György út 84/b

Postafiók 571
Budapest 1539

FELADÓ: **Feladáskor kérjük bérmentesíteni!**

Név: ....

Cím: ....

Helység: ....

Irányítószám: ....

Telefon: ....

☐ A hirdetés egyéni és egyedi jellegű, ezért kérem ingyenes megjelentetését. Kijelentem, hogy annak tartalma nem sérti senki szerzői jogát.

☐ A hirdetés kereskedelmi célt szolgál. Mellékelem a soronként (60 karakterenként) 300 forintnak megfelelő összeg átutalásáról az igazoló szelvény másolatát.
A címzett: Új Alaplap, 1539 Budapest, Pf. 571, illetve átutalásnál az OTP 11706016-20788599 számlaszámra.

Bélyeg helye

Új Alaplap

VI., Dózsa György út 84/b

Postafiók 571
Budapest 1539

Az Új Alaplap lemezmellékletének tartalma **1997. SZEPTEMBER**

IBM Network Station: jelen és jövő

Az IBM egyike azon néhány cégnek, amelyeknek valódi Network Computer-történetük (is) van, hiszen a Kék Óriás néhány hónapja tömegesen szállít ilyeneket a piacra. Július végén pedig a Network Station (NS) nevű IBM NC-k ún. 100-as sorozatát követően megjelentek a 300-asok is. Az NS 300-at szintén az IBM Network Computing Devices Inc. vállalat gyártja. Ezek a gépek a 100-as szériától az órajelfrekvenciában és a RAM méretében különböznek: 66 MHz, illetve 16 Mbájt, összehasonlítva a 100-asok 33 MHz-es, illetve 8 MHz-es értékeivel. A szoftver ugyanaz mindkét sorozatnál: az IBM saját „lecsupaszított" operációs rendszere; Java virtuális gép; natív emulátorok az IBM 3270-es és 5250-es terminálokhoz; az NCD cég WinCenter Pro szoftvere, amely az NT többfelhasználós változata; a Spyglass cég vagy a Navio Communications cég böngészője. Az NS újabb sorozata az 1000-es lesz, amely várhatóan 1997 IV. negyedévében jelenik meg, és ez lesz az első PowerPC 603 típusú processzoros Network Station. Az órajele várhatóan 200 és 350 MHz közötti, a RAM mérete pedig 32 Mbájt lesz.

Hálózati vírusveszély

A Computer Associates cég Cheyenne divíziója szponzorálta a National Computer Security Association (NCSA) által végzett kutatást, amely a vállalati hálózatokat 1997-ben fenyegető vírusveszélyekkel volt kapcsolatos. Az eredmények azt mutatják, hogy a vírusfertőzések száma 1996-ban megduplázódott, és a hálózatba kötött PC-k 40%-át valamilyen formában vírusok támadják, támadhatják meg. A felmérés során a megkérdezettek 99%-a már talált vírust rendszerében, 71%-uk alkalmaz valamilyen eljárást a vírusprobléma kezelésére, 43%-uk pedig úgy érzi, hogy a számítógépes vírusokkal kapcsolatban az utóbbi időben nőttek a gondok. A tanulmány szerint minden tíz vírusfertőzésből hat még mindig floppylemezen érkezik, ugyanakkor az Interneten keresztül „beszerzett" vírusok száma egyre növekszik. A válaszadók 45%-a válaszolt úgy, hogy legutóbbi vírusproblémát fertőzött állomány letöltése (19%) vagy e-mailhez csatolt fertőzött anyag (attachement) okozta (26%). Amikor vírusok támadják meg a hálózatot, a kár leginkább a tárolt adatokat éri, csökken továbbá az alkalmazásások megbízhatósága, és gyakoribbak a rendszerösszeomlások. A Word Macro Virus a tanulmányba bevont cégek 49%-ánál okozott fertőzést. A Word Concept Virus „tevékenysége" 1997-ben megháromszorozódott. A makróvírusok széles körben terjednek, mert bármely olyan dokumentumállományhoz, amely makrókódot tartalmaz, hozzáfűzhetik magukat. (Lásd erről részletesen Vírusőrjárat rovatunkat a 38. oldalon.) Az NCSA értékelése szerint a Cheyenne InocuLAN programterméke az összes vírus több mint 95%-át azonosítani és hatástalanítani tudja.

Vezeték nélküli LAN-ok szabványa

Az IEEE nemzetközi szabványosítási szervezet Standards Activity Boardja ratifikálta a 802.11-es számú szabványt, amely a vezeték nélküli lokális hálózatokra vonatkozik. A szabvány nem kevesebb, mint hét évig készült, és rövid időn belül egész sor „zsinór nélküli" egységes LAN termék megjelenését fogja eredményezni. A 802.11-es szabvány a vezeték nélküli LAN-ok fizikai rétegét és Media Access protokolljait határozza meg. A fizikai réteg definíciója három átviteli opciót ismer: infravöröset és két rádiófrekvenciást (Direct Sequence Spread Spectrum és Frequency Happing Spread Spectrum). A kijelölt 5 GHz tartományban az adatok továbbítása 10-20 Mbit/s sebességgel történhet. Szakértők szerint az IEEE szabvány megjelenésével csökkeni fog a vezeték nélküli LAN-ok portonkénti ára, amely jelenleg magasabb, mint a fizikai érpáras hálózatoké.

Bay Networks: kapcsolók árszintje

Az amerikai Bay Networks vezető hálózati eszközgyártó cég új Ethernet/Fast Ethernet-alapú munkacsoport-kapcsolóival és 10Base-T bázisú rétegelhető hub eszközeivel a szakértők szerint új árszinteket jelöl ki az IT iparágban. A BayStack Ethernet Desktop Switch, amely 24 darab 10 Mbit/s-os porttal és 11 darab 100 Mbit/s-os porttal rendelkezik, 94 dolláros portonkénti árral jelenik meg a tengerentúli piacon. A BayStack 304 Ethernet Segment Switchnek, amely 12 darab 10 Mbit/s-os és 11 darab 100 Mbit/s-os portot tartalmaz, portonkénti ára 125 dollár. Az új BayStack 150 Series típusú 40Base-T hubok felügyeleti képességekkel is rendelkeznek; a portonkénti ár ezeknél 43 USD.

Cisco: fiókirodai router

Leginkább a fiókirodákkal rendelkező cégeknek ajánlható a Cisco 3620 Router ISDN útválasztó eszköze. Fő jellemzője: flexibilitás, modularitás, nagy teljesítmény. A berendezés kitűnő remote-access-server (RAS) megoldást nyújt. Szakértők szerint ez lehet a sikeres Cisco 2500-as sorozat folytatása a következő tíz évben. Van benne két Ethernet port, nyolc ISDN BRI port és egy egycsatornás T1 interfészcsatlakozás (beépített CSU, azaz csatornaszerviz egységgel). Mint más nagyobb teljesítményű Cisco termékeknél, a 3620-asnál is specifikálni kell mind a kártyaszámokat, mind az interfészszámozást a konfigurációs utasításokban. A Cisco a jelek szerint nem kívánja kiszorítani 2500-as routereit a 3620 típusú útvonalválasztókkal, a 2500-asok alacsonyabb áruk miatt is még hosszú ideig piacon maradhatnak. A 3620-as router, bár 50-100 százalékkal többe kerül, mint a vele hasonlóan konfigurált 2500-as típus, várhatóan a 2500-asnál nagyobb életciklust fog megérni olyan hálózatoknál, amelyekben a növekedés és a változtatás rendszeres követelmény.

3Com a kis irodákban

Kombinált hang- és adatátviteli hálózati megoldások létrehozására kötött szövetséget az amerikai 3Com és a német Siemens. Ennek nyomán a Siemens hangkapcsolói (HiCom PBX-ek) a 3Com Fast IP Layer3 kapcsolási technológiájával (CoreBuilder, NetBuilder) és TrancendWare szoftverével fognak együttműködni. Így a kliensek bizonyos típusú forgalmi helyzeteknek megfelelő szolgáltatási osztályokat vehetnek igénybe. Ugyancsak újdonság: kis irodák és fiókok igényeihez tervezett, az OfficeConnect platformján alapuló új termékeket vezetett be a 3Com a távoli hozzáférés, a vállalati útválasztók és a koncentrátorok termékcsoportjában. Az új eszközök: OfficeConnect Remote Access Server 1000, OfficeConnect NETBuilder vállalati útválasztók, valamint OfficeConnectHub TP16C. Az OfficeConnect Remote Access Server 1000 megoldást jelent azoknak, akik lehetővé akarják tenni gyakran úton lévő vagy és távolabb dolgozó munkatársaik számára a vállalati LAN elérését. Az OfficeConnect NETBuilder routereket a vállalati intranetek kiterjesztésére tervezték. Az OfficeConnect Hub TP16C egy max. 16 portos 10BaseT koncentrátor.

Unicomp: RND-disztribúció

Július elsejétől a Lanex mellett új disztribútora van Magyarországon az izraeli RND Network cégnek (amely nem tévesztendő össze az ugyancsak izraeli RAD csoporttal). A székesfehérvári Unicomp Kft. most megszerzett disztribúciós joga Magyarországon kívül Romániára is kiterjed, az RND véleménye szerint ugyanis jelentős bővülésre és az értékesítési csatornák kiszélesítésére kell felkészülni mindkét országban. Az RND routerei a SOHO piac igényeit hivatottak kiszolgálni, a switchek pedig fix 10 Mbit/s-os illetve 100 Mbit/s-os eszközök. Az Unicomp az RND routerek és switchek révén képes kielégíteni a jelenleg „átlagosnak" tekinthető hálózati igényeket, ezek mellett azonban egy sor olyan termék is a magyar (és román) piacra kerül, amelyek az elmúlt évek fejlesztési eredményeit reprezentálják. Ilyen például a Data Communications szaklap Hot Product díjával elismert Vgate virtuális router vagy a PC Magazine szerkesztői által ajánlott Web Server Director. A Unicomp által forgalmazott RND eszközökkel a közeljövőben egy roadshow keretében közelebbről is megismerkedhetnek az érdeklődők.

Kovács Attila

GroupWise 5

Csoportról csoportra

A cégek számára gyakran jelent problémát a munkacsoportok információcseréjének szervezetlensége, az integrált infrastruktúra hiánya. A Novell csoportmunka-támogató programja, a GroupWise 5 erre a problémára kínál megoldást, az adott cégen belül szükséges munkaszervezést azonban nem helyettesíti.

Az itthon (MHB, Parlament) és külföldön (American Airlines, Borland, CBS News, FBI stb.) egyaránt jó nevű referenciákkal rendelkező termék helyes alkalmazása esetén számos vállalati problémára nyújthat megoldást.

Talán mindenütt a legfőbb gond a sokféle információs rendszer használata, és az általuk kezelt, illetve tárolt adatok elérhetősége. A probléma megoldását olyan információs programcsomag jelenti, amely a különböző utakon érkező információt egy strukturált adatbázisban tárolja, és könnyen kezelhető felületen teszi elérhetővé.

A GroupWise az elektronikus levél, a hangposta, a fax, az irodán belüli levelezés és a naptár egyesítésével nyújt segítséget, és formálja áttekinthetővé a kommunikációt. A lényeg az, hogy az integráció lehetővé teszi a szállító közegtől történő függetlenedést, és elősegíti a tartalomra való koncentrálást. A GroupWise erénye az állapot- és nyomkövetés, melynek alkalmazásával a munkacsoportokon belüli folyamatok könnyen optimalizálhatók. A mindenkor (még távoli felhasználás esetén is) biztonságos és titkos kommunikáció segítségével könnyen megoldhatók az elektronikus levelezést kiváltó egyéb szolgáltatások is, például a tárgyalások szervezése különböző megosztott naptári rendszereken keresztül, illetve a kiértesítés.

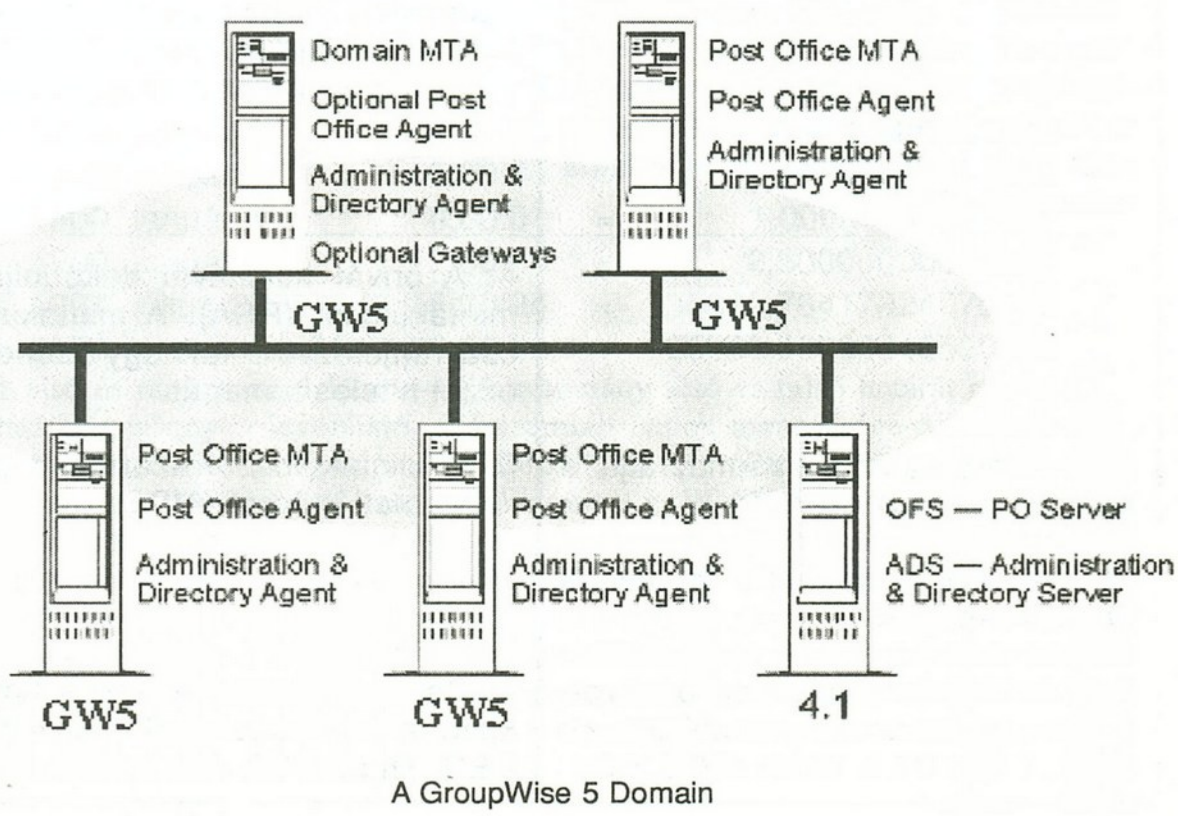
A Busy Search nevű szolgáltatás segítségével végignézhetjük minden érintett felhasználó és erőforrás naptárát egy-egy időpont kiválasztásához. Itt vetődik fel az a kérdés, hogy hogyan oldható meg az elektronikus levelezésen kívüli dokumentum vagy fájlcsere is, hiszen az e-mail számos hátránnyal rendelkezik: nincs rendes verziókövetés, rossz a tárolási módszer. GroupWise környezetben a dokumentumszolgáltatások segítségével megoldható, hogy a felhasználó csak bizonyos hivatkozásokat kapjon meg, illetve minden dokumentumból csak egy példány létezzen — ügyelve emellett arra, hogy egy dokumentumhoz egyszerre több felhasználó is hozzáférhessen.

A dokumentáció-kezelés körébe tartozik a weboldalak és webhivatkozások (URL-ek) kezelése is, mivel a referenciák azonnali hozzáférést biztosítanak a Webhez. Kényelmes megoldás továbbá a távoli felhasználók internetes hozzáférési lehetősége egy webböngészővel. Ennek keretében a megfelelő jogosultsággal rendelkező személy számára minden GroupWise szolgáltatás elérhető a böngészővel, függetlenül a GroupWise által támogatott környezettől (platformtól). A GroupWise-alkalmazások szolgáltatásai emellett nemcsak a GroupWise klienssel vagy a hálóböngészővel, hanem bármilyen MAPI 1.0 alkalmazással is képesek együttműködni.

A manapság elterjedt telefonközpontok kiváltására alkalmas a számítógép—telefon integráció, amelynek segítségével a felhasználók közötti telefonálás bármilyen TAPI vagy TSAPI telefonkapcsolaton keresztül megvalósulhat, a hívó fél azonosításával együtt. Sok helyütt már elterjedt szolgáltatás a telefonos hozzáférés a GroupWise információkhoz. Ennek segítségével bármilyen tone üzemmódú készülékről titkos kódok segítségével hozzáférhetünk a szolgáltatásokhoz. A szövegnek beszéddé alakításával levelek, események és feladatok felolvasása is lehetővé válik a felhasználók számára.

Tekintettel a meglévő NetWare adminisztrátorokra, a GroupWise 5 is ugyanazt az eszközt használja, mint a NetWare. Az ügyfél/kiszolgáló kapcsolaton át nő a rendszer biztonsága és megbízhatósága, amit tovább javít az, hogy minden adat kódolt formában kerül átvitelre, valamint az adatbázisok képesek önjavításra is. (Az adminisztrátort levélben értesíti a kezelőrendszer a hiba felléptéről és javításáról.) A GroupWise nyíltsága nagymértékben elősegíti az integrálhatóságot az SMTP/MIME, SNMP (ManageWise), NDS (X.500) és X.400 rendszerek támogatásával, valamint más átjárókkal az elterjedt levelezőrendszerek számára.

Varga János



A GroupWise 5 Domain

Elektronikus kereskedelem

Biztonsági protokoll

Az Új Alaplap előző számában a hónap témája már kitért a hitelesítés és az azonosítás kérdéseire. Akkor az elektronikus kereskedelembe nem akartunk belemélyedni, mert a későbbiekben készítünk majd arról is egy tematikus összeállítást. Ugyanakkor a kérdés annyira „benne van a levegőben", hogy inkább már most közreadjuk, és nem „jegeljük" tovább szerzőnk írását, amely az általános célú hitelesítési protokollokkal és a pénzügyi tranzakciók védelmére kifejlesztett speciális protokollokkal foglalkozik.

A gyakorlatban a hitelesítési feladatot egy kriptográfiai protokoll valósítja meg, amely sokszor rejtjelezéssel is védi az érzékeny információkat. A protokoll elvileg az OSI hálózati rétegmodell több szintjén is dolgozhat, a nyílt hálózati környezetben azonban alapvetően különböző két megközelítés bizonyult hatékonynak; a hálózati szintű és az alkalmazásszintű hitelesítés.

Secure Socket Layer

A Netscape által kifejlesztett Secure Socket Layer protokoll az egyik legelterjedtebb megoldás a hálózat biztonságosabbá tételére. A protokoll hálózati szinten dolgozik, vagyis az OSI rétegmodell szerint a szállítási réteg szintjén avatkozik be az adatátvitelbe. Ennek megfelelően az adatcsomagokat hitelesíti és rejtjelezi, és transzparens módon elvileg bármilyen alkalmazás futhat fölötte. A gyakorlatban azonban különkülön portcímeket foglalnak le az ezáltal biztonságossá tett alkalmazásoknak, ami könnyebbé teszi a csomagszűrő típusú tűzfalak munkáját. A teljes kapcsolatot védi, viszont nem lát rá az applikációs szinre, ezért egy fájl hitelesítése ezen a módon nem tehető meg. Legújabb verziója a 3.0-s.

A protokollt úgy tervezték meg, hogy védett legyen az ún. közbeékelődő (man-in-the-middle) típusú beavatkozás ellen. Ennek az interaktív protokollok elleni általános támadási módszernek a lényege, hogy amikor Aladár és Béla fel akarja venni a kapcsolatot, akkor Viktor közéjük áll, és így mindkét gyanútlan áldozat Viktorral veszi fel a kapcsolatot, aki miközben fordít Aladár-Viktor nyelvéről Viktor-Béla nyelvére, megismeri a védendő adatokat. Emiatt súlyos nehézségek merülnek fel, amikor egy proxy szerveren keresztül folyik a kapcsolat, hiszen egy proxy éppen ezt teszi: fordít a külvilág és a lokális hálózat között.

A fenti probléma megoldható az úgynevezett SSL tunnelling (alagút) protokollal. A folyamat nagy vonalakban úgy játszódik le, hogy a szerver bizonyítja önazonosságát a kliens előtt, majd egy opcionális fázisban a hitelesítés fordítva is megtörténhet. A további lépések a lenti táblázatban láthatók.

Kliens	Szerver
1. Ki vagy?	
	2. X. Y. vagyok, itt a nyilvános kulcsomhoz tartozó tanúsítvány, és a következő algoritmusokkal tudok rejtjelezni: (lista).
3. Generál egy véletlen mintakulcsot. Rejtjelezi a szerver nyilvános kulcsával (RSA).	
	4. A privát kulcsával dekódolja a mintakulcsot (RSA). A mintakulcscsal rejtjelezve elküld egy üzenetet, ezzel hitelesíti magát.
5. A mintakulcsból származtatja a kapcsolati kulcsot (MD5).	5. A mintakulcsból származtatja a kapcsolati kulcsot (MD5).
6. Opcionális: a kliens bizonyítja önazonosságát.	
7. Csomagok rejtjelezése a következő algoritmusok valamelyikével: DES, Triple DES, IDEA, RC2, RC4.	7. Csomagok rejtjelezése a következő algoritmusok valamelyikével: DES, Triple DES, IDEA, RC2, RC4.

Pénzügyi tranzakciók védelme

Az Interneten keresztül lebonyolítható kereskedelmi és pénzügyi műveletek terjedésének sokáig az volt a legfőbb gátja, hogy nem voltak erre a célra általánosan elfogadott és megfelelő biztonsági szintű rendszerek. Időközben nagyon sok ilyen protokollt javasoltak, és megtörténtek az első lépések az egységesítés felé is. Ennek legfontosabb jele, hogy a két legnagyobb hitelkártya-kibocsátó cég, a Visa és a Mastercard megegyezett a Microsofttal egy közös protokollban.

Ezek az eljárások egyrészt a már többé-kevésbé globális hitelkártyarendszerekre épülnek. Másrészt pedig a legtöbb rendszerben megvannak a nyilvános kulcsú kriptográfia elemei. Ez azt jelenti, hogy a rendszer működése feltételezi a háttérben felépített, nyilvános kulcsú titkosítási infrastruktúrát. Első közelítésben ez nem más, mint egy telefonkönyv, amely tartalmazza a nyilvános kulcselemeket, azon szabályok összességét, amelyek alapján egy új rekord felvehető, egy régi törölhető, illetve módosítható. A központ (illetve egymással együttműködő több központ) minden résztvevőnek egy tanúsítványt (certificate) ad, amely bizonyítja, hogy az adott résztvevőnek valóban az a nyilvános kulcsa, amit ő állít. Technikailag ezt úgy hajtják végre, hogy a résztvevő nyilvános kulcsát, nevét és néhány adminisztratív adatát a központ titkos kulcsával digitálisan aláírják. Ezután a rendszerben mindenki ellenőrizni tudja a tanúsítványokat, mert a központ nyilvános kulcsa mindenki számára elérhető. A rendszernek rendelkeznie kell

— a fentieket lehetővé tevő technikai háttérrel, és

— megfelelő jogi szabályozással, amely jogi eszközökkel védi a rendszer elemeit.

Ezek után is probléma marad azonban, hogy a fentiek szervezése országos szinten folyik, míg az Internet nemzeteken átnyúló háló. Másik probléma, hogy a nyilvános kulcsú műveletek rendkívül lassúak, egy-két nagyságrenddel lassúbbak, mint más kriptográfiai alapeszközök. Költséges továbbá a kibocsátó központtal (bankkal) fenntartott online kapcsolat is. Ez megjelenik a rendszer költségeiben, amelyet részben a felhasználó visel ugyan, de ez oda vezet, hogy ilyen tranzakciókat csak bizonyos összeghatár felett végeznek. Ezért a fejlődés következő logikus lépése olyan rendszerek kifejlesztése, amelyek talán nem garantálnak olyan magas szintű biztonságot, mint a korábban említettek, de olcsóságuk, hatékonyságuk miatt a mindennapok aprócseprő vásárlásainál is használhatók. Ilyen például a Rivest és Shamir által kidolgozott Payword nevű rendszer is.

Payword

A protokoll zsetonok tranzakcióját kezeli, egy zseton értéke lehet 1 Ft, de akár 1000 Ft is, a rendszer igényeitől függően. A protokoll a következő lépésekből áll:

— Az ügyfél létrehoz egy számlát a banknál, amely tanúsítványt ad az ügyfélnek, feljogosítva az ügyfelet (digitális) zsetonok létrehozására.

— Az ügyfél offline módon legyárt egy adag zsetont, mielőtt a kereskedővel kapcsolatba lép.

— A zsetonok egy hash függvény segítségével láncszerűen összekapcsolódnak, és az ügyfél az egész láncot hitelesíti digitális aláírásával.

— A kereskedő beváltja a zsetonokat a banknál.

A zsetontechnika

1. Az ügyfél készít n darab zsetont (zs_0 zs_1 zs_2... zs_n) a következőképpen. Generál egy véletlen számot, ez lesz zs_n, az n-edik zseton, majd az i-edik zseton az i+1-edik hash függvény szerinti képe lesz: $zs_i = h(zs_{i+1})$.

2. Az ügyfél kap a banktól egy tanúsítványt, amely tartalmazza a bank nevét, az ügyfél nevét, nyilvános kulcsát stb. a bank titkos kulcsával digitálisan aláírva.

3. Amikor az ügyfél először lép kapcsolatba egy kereskedővel, akkor elküld neki egy kötelezvényt, amely tartalmazza a kereskedő nevét, az ügyfél tanúsítványát, a nulladik zseton értékét, s egyéb információkat az ügyfél titkos kulcsával aláírva. Ennek alapján a bank nyilvános kulcsa segítségével a kereskedő ellenőrizni tudja, hogy az ügyfél–bank kapcsolat valóságos-e.

4. Az ügyfél úgy fizet például 10 zsetont a kereskedőnek, hogy elküldi a 10. zseton értékét, s mellé a 10-es sorszámot (zs_{10}, 10). A következő fizetésnél ismét elküldi a 10+k-adik zseton értékét és a 10+k sorszámot. A kereskedőnek elég azt tárolnia, hogy az utolsó fizetésnél melyik zsetont kapta, s annak mi volt a sorszáma. Magyarán azt, hogy hány zsetont kapott eddig összesen, s mennyi az utolsó zseton értéke.

5. A kereskedő időnként elküldi a banknak az ügyfél nevét, kötelezvényét és az utolsó zseton értékét, indexét (k). A bank ellenőrzi, hogy az ügyfél valóban k zsetont költött-e, hiszen a zs_k-ra k-szor alkalmazva a hash függvényt, a kötelezvényben lévő zs_0-t kell kapni, majd elvégzi az átutalást.

Biztonság, hatékonyság

Azzal, hogy a kötelezvényben a kereskedő és az ügyfél neve is szerepel, a zsetonok kereskedőspecifikusak és ügyfélspecifikusak, más kereskedő számára értéktelenek. A kereskedő a hash függvény tulajdonságai miatt nem tud magának zsetont gyártani, hiszen ehhez fel kellene törnie a hash függvényt, azaz „visszafelé" kellene lépnie a képzési láncban. Ugyanakkor a fizetési rendszer nem tartalmazza, hogy mit vettünk (vagy vettünk-e egyáltalán valamit), tehát ez a rendszerben bizalmi kérdés.

A protokoll rendkívül hatékony abban az értelemben, hogy a szükséges online kapcsolatok száma minimális. A kereskedőnek nem kell kapcsolatba lépnie a bankkal, sem az első, sem a további vásárláskor. Elegendő mondjuk havonta elküldeni az utolsó zsetont a hozzá tartozó adatokkal, de ezt sem kell online módon megtenni.

A hitelesítési feladatok megoldása nyílt hálózatokban az egyik legnagyobb kihívás az Internet fejlődésében. Komplex megoldása nemcsak kriptográfiai módszerek, szabványok létrehozását kívánja meg, hanem egy nyilvános kulcsú infrastruktúra létrehozását is, annak törvényi, szervezeti garanciáival együtt. Ugyanakkor szerintünk egy-egy problémára létezhetnek egyszerűbb megoldások, amelyek jól szolgálhatnak az átmeneti időszakban, s egyfajta lépésenkénti, szerves fejlődés alapjait képezhetik.

Papp Pál

Hivatali idő: 2000

Az augusztusi szám kiemelt témája érintette a 2000. évvel kapcsolatos informatikai problémákat is. Most ennek egy hétköznapi vetületéről ejtünk néhány szót. Például az Office programok időpontokat tárolnak, vizsgálják őket, számításokat végeznek velük. Biztonságosak-e a legelterjedtebb irodai szoftverek, a WindWord, az Excel, a Works, a Lotus 1-2-3 és társaik a kritikus évbe, 2000-be való átlépés után? Ez azon múlik, hogyan tárolják a dátumok, időpontok adatait. (Szerencsére egy kaptafára megy! Igaz, a részletekben vannak kisebb-nagyobb különbségek.)

Az alapgondolat az, hogy egy — tulajdonképpen önkényes — dátumtól kezdve folyamatosan kell számlálni a napokat. (Tehát az éveket, hónapokat is átszámolják napokba.) Ha a nap tört részeivel is dolgozunk (órák, percek, másodpercek), egyetlen valós számban összefoghatjuk a napokat és a napok törtrészeit. Excel 5-ös példák:

1900. január 0.	0:00:00	=	0,00000000000000
1900. január 1.	0:00:00	=	1,00000000000000
1900. január 1.	0:00:01	=	1,00001157407407
1900. január 2.	12:00:00	=	2,50000000000000
1997. július 28.	18:20:25	=	35639,76417824070000
2078. december 31.	23:59:59	=	65380,99998842590000

(Az első és utolsó sor a dátum—idő tartomány alsó és felső határát mutatja.)

A tanulság könnyen levonható. A leszámlált napok mennyisége nem érzékeny az ezredfordulóra, észre sem veszi, mikor lépi át. Igaz, az említett önkényes kezdő időpont nem azonos a különféle platformok (Windows 3.x, Win95, Windows NT, MacIntosh System 7.x) és különféle programverziók esetében, és ebből sejthetően a háttértámogatást adó WordBasic, Visual Basic programokban sem. De az mindegyikre igaz, hogy a számlált napokkal nyilvántartott dátum azonos módon, észrevétlenül halad át a bűvös kétezresen. Sőt, még a DOS fájl- és könyvtárdátumainak évei is folyamatosan fedik le az 1980. és 2099. közötti időszakot. A megfogalmazás nem stilisztikai okból árnyalt: a DOS nem valamilyen azonos egységre átszámított értéket, hanem bithatárra pakolt, de különálló év, hónap stb. adatot rögzít.

Segesdy Gábor

A makró kitűnő táptalaj

Az Office 97 és a vírusok

A Microsoft 1997 elején dobta piacra Office programcsaládjának legújabb változatát, amely elődeihez képest számos vonatkozásban lényegesen átalakult. A csillogóbbá varázsolt külső alá betekintve a legkomolyabb változást az jelentette, hogy az addig enyhén szólva is eklektikus képet mutató makróprogramozást végre egységesítették. Vajon milyen változásokat hozott ez a napjainkban első számú közellenségnek számító makróvírusok szempontjából?

Történelmi okokra visszavezethetően az Office korábbi változataiban két, egymástól nagyon különböző makrónyelv szerepelt. Az egyik a Word 2.0-ból a Word 6.0/7.0-ra örökölt WordBasic, a másik pedig az Excel és az Access számára bevezetett VBA (Visual Basic for Applications). Az előbbi nem más, mint egy nagyon egyszerű Basic nyelv, megtoldva a párbeszédablakok könnyített kezelésével és a Word különféle műveleteit meghívó mintegy 800 beépített függvénnyel. Ezzel szemben a VBA jóval komolyabb, az objektumorientáltság felé közelítő, szintén Basic-alapú programozási környezet.

Az utóbbi egy év folyamán a makróvírusok számítottak a legelterjedtebbnek a számítógépes kártevők körében, ezért felvetődik a kérdés, hogy vajon ebben a gyökeresen megváltozott helyzetben mihez kezdenek a Word 6.0/7.0 alatt kényelmesen éldegélő vírusok.

Beépített vírusvédelem

Már a Word 7.0 javított változatában, a 7.0a-ban is megjelent az úgynevezett vírusvédelmi opció, ezt azután minden hibájával együtt megtartották a Word97-ben is. Addig rágták a Microsoft technikusainak a fülét a felhasználók és a vírusvédelmi szakemberek, amíg végre hajlandóak voltak tudomást venni a makróvírusokról, és beépítettek egy figyelmeztető mechanizmust — a rossz nyelvek szerint amúgy Microsoft-módra, azaz a valódi problémák és elvárások figyelmen kívül hagyásával, csak a maguk feje után haladva.

A makróvírusok (néhány patologikus eset kivételével) kétféle mechanizmus segítségével terjednek: az automatikusan végrehajtódó makrók, illetve a Word parancsait felüldefiniáló makrók segítségével. A vírusvédelemtől legalább azt el lehetne várni, hogy figyelmeztesse a felhasználót, ha a megnyitandó dokumentum bármiféle makrót tartalmaz, továbbá hogy ki lehessen kapcsolni az automatikus makróvégrehajtást és a Word-parancsok átdefiniálását. Ezzel szinte az összes vírus alól ki lehetett volna húzni a képzeletbeli szőnyeget. Hogy ehelyett mit kaptunk? Az 1. ábrán bemutatott figyelmeztető ablakot.

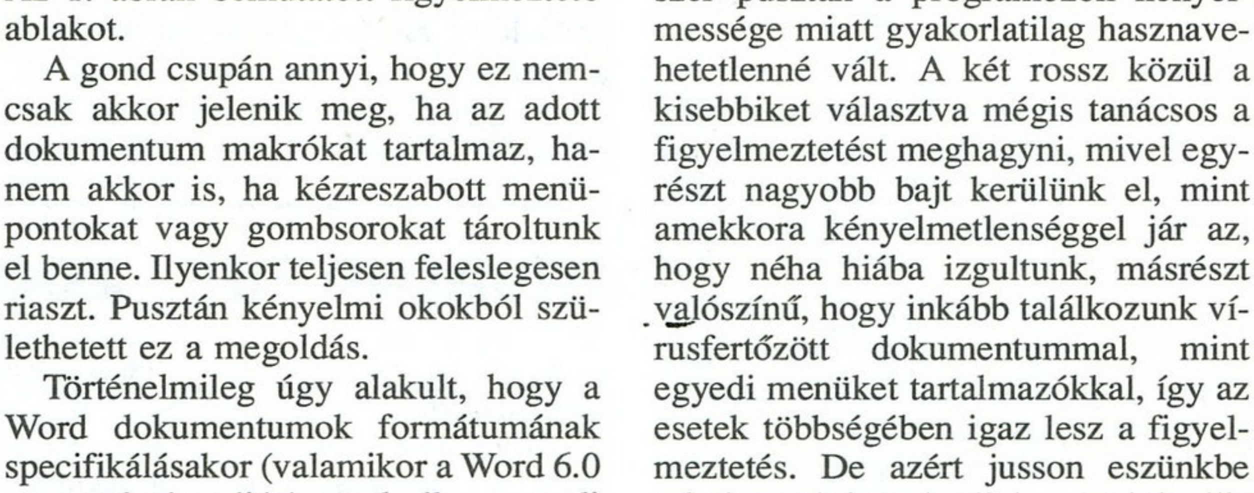
A gond csupán annyi, hogy ez nemcsak akkor jelenik meg, ha az adott dokumentum makrókat tartalmaz, hanem akkor is, ha kézreszabott menüpontokat vagy gombsorokat tároltunk el benne. Ilyenkor teljesen feleslegesen riaszt. Pusztán kényelmi okokból születhetett ez a megoldás.

Történelmileg úgy alakult, hogy a Word dokumentumok formátumának specifikálásakor (valamikor a Word 6.0 megszületése táján) a makrókra, egyedi kialakítású menükre és gombsorokra vonatkozó információkat egyetlen struktúrában tárolták. Akkor még senki sem gondolta, hogy a makrók ennyire népszerűek lesznek, ezért dobták egy kalapba ezeket az egymástól meglehetősen távol eső elemeket. Amikor a „vírusvédelmet" beépítették, kényelmi szempontból úgy döntöttek, hogy csak ennek a struktúrának a meglétét vizsgálják, azt már nem, hogy makrók is vannak-e benne. A Wordbe beépített mechanizmus tehát olyankor is vírusgyanú miatt sikongat, amikor teljesen ártalmatlan dokumentumot nyitunk meg. Márpedig általános tapasztalat (nemcsak a víruskeresők esetében, hanem mindenféle diagnosztikai rendszernél is), hogy minél több a téves riasztás, annál inkább romlik a felhasználók körében a rendszer hitele.

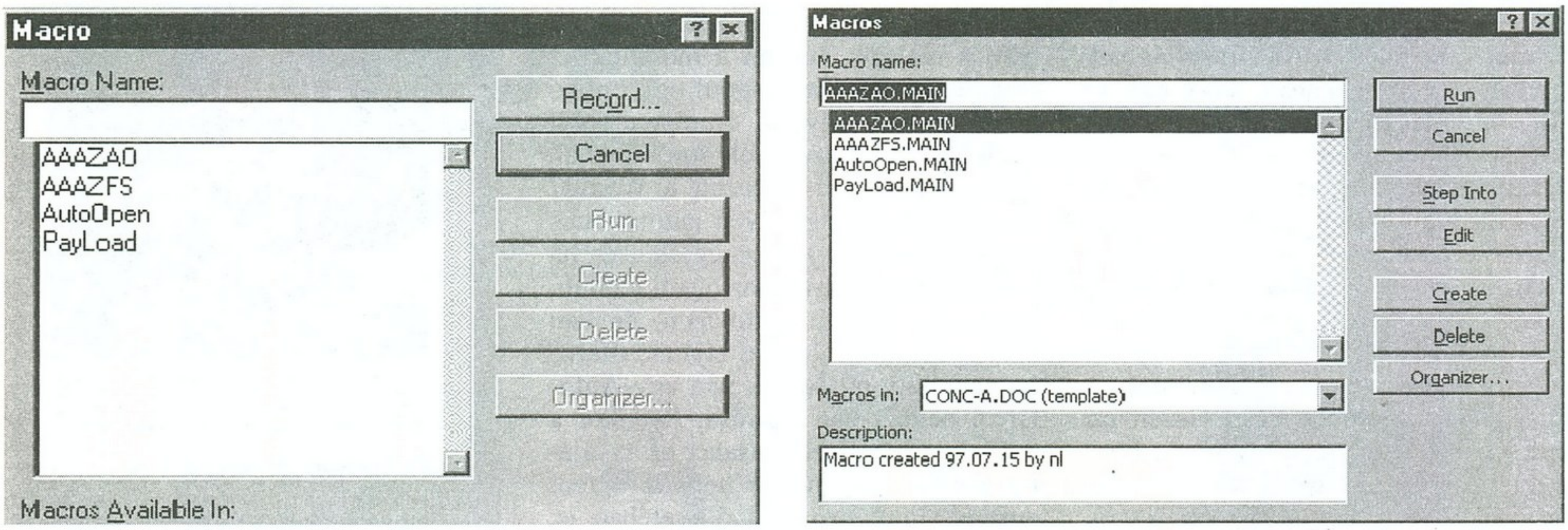
A végeredmény az, hogy egy idő után a felhasználó már automatikusan engedélyezi a makrókat, meg sem vizsgálva a dokumentumot közelebbről, vagy esetleg teljesen ki is kapcsolja az ellenőrzést. Magyarán a vírusvédelmi rendszer pusztán a programozók kényelmessége miatt gyakorlatilag használhatatlanná vált. A két rossz közül a kisebbiket választva mégis tanácsos a figyelmeztetést meghagyni, mivel egyrészt nagyobb bajt kerülünk el, mint amekkora kényelmetlenséggel jár az, hogy néha hiába izgultunk, másrészt valószínű, hogy inkább találkozunk vírusfertőzött dokumentummal, mint egyedi menüket tartalmazókkal, így az esetek többségében igaz lesz a figyelmeztetés. De azért jusson eszünkbe minden vakriasztásnál, hogy minimális plusz időráfordítással rendesen is meg lehetett volna oldani ezt a vírusvédelmet.

Automatikus víruskonverzió

A kompatibilitás érdekében a korábbi Word változatokban megírt makrókat gond nélkül át lehet konvertálni VBA nyelvre. Ez a gyakorlatban azt jelenti,



1. ábra



2. ábra

hogy egy Word 6.0/7.0 dokumentum megnyitásakor a Word a benne levő makrókat automatikusan Visual Basic nyelvre ülteti át.

Itt rögtön egy problémával kellett megküzdeniük a Microsoft szoftvermérnökeinek. A WordBasickel szemben a VBA objektumorientált nyelv, minden végrehajtandó eljárásnak valamilyen objektumhoz kell kapcsolódnia. Az azonos célokat szolgáló eljárások közös objektum alá kerültek, logikai kapcsolatok alapján rendeződve. A WordBasicben az eljárások csak egy nagy kupacba lettek dobálva, semmi strukturáltság nem volt a nyelvben. Továbbá nemcsak a makróprogramozást reformálták meg a Word97 esetében, hanem a szövegszerkesztő felépítését is. Ezért az egyes eljárások nem pontosan azt csinálják, amit korábbi változatbeli elődeik.

Az egyik megoldás az lehetett volna, hogy a Microsoft programozói minden WordBasic utasításhoz megtalálják azt az objektumot és eljárást, amely ugyanazt az akciót megvalósítja. Ha ezt a megoldást választják, akkor a konverzió során kapott program nem pontosan ugyanazt csinálta volna, mint elődje (hiszen a funkciók helyenként jelentősen megváltoztak), hanem néha váratlan hibákat eredményezett volna. A megoldás ehelyett az lett, hogy (hangsúlyozottan ideiglenesen) a teljes WordBasicet beemelték a VBA-ba. Létrehoztak csakis a konverzió kedvéért egy WordBasic nevű objektumot, és a régi WordBasic összes függvényét és eljárását bepakolták ezen objektum eljárásaiként. Így a konvertált kód pontosan azt csinálja, amit az eredeti, és a felhasználó majd tetszése szerint átírja azt valódi VBA objektumokat felhasználó, natív kódra. Még mindenkinek van néhány éve erre az átírásra, mert bár a Microsoft figyelmeztet, hogy a kompatibilis WordBasic nevű objektumot nem fogja örök időkig benne hagyni a VBA-ban, a történelem megtanított minket arra, hogy a régebbi verziókkal kapcsolatot tartó, ideiglenesnek titulált megoldások igen hosszú életűek tudnak lenni.

Mindenesetre az automatikus konverzió egyik első számú haszonélvezői a makróvírusok. Természetükből adódóan ezek a vírusok röghöz kötöttek: csak abban a szövegfejlesztői környezetben élnek, amelyben megírták őket (nincs átjárás, mert a WordPerfect például nem kezeli a Word makróit és fordítva). Egy annyira új környezetben, mint a VBA, esélyük sem lett volna a túlélésre — de a WordBasic-kompatibilitási objektum révén sikerült mentőövet dobni elébük. Ennek révén — mint gyakorlatilag minden WordBasic makró — a vírusok is zökkenőmentesen lefutnak VBA alatt. Külsőre egy kicsit másképp fognak kinézni, de lefelé kompatibilisak és emiatt ugyanazt csinálják. (Lásd a két bekeretezett programrészt.)

A Concept evolúcióját, a Word 7.0 illetve Word97 alatti megjelenését a 2. ábra szemlélteti, annak alapján, hogy mit látunk a megszokott Tools|Macro paranccsal. A .MAIN toldalék azért szerepel, mert a Word97 makró fogalma kicsit tágabb: makrónak kell tekinteni minden modulban a Public kulcsszóval deklarált eljárásokat és a MAIN nevű eljárást. A Word 6.0 makrókban csak ez utóbbi van, ezt jelzi a .MAIN kiterjesztés.

Bármennyire is különbözőnek tűnik, a Word 7.0 illetve Word97 alatti két vírus ugyanaz, legalábbis abban az értelemben, hogy ha egy hagyományos Concept vírussal fertőzött dokumentumot megnyitunk Word97-tel, akkor az automatikusan a WM97.Concept vírussá konvertálódik, és a Word97 világá-

```
For i = 1 To iMacroCount
        If MacroName$(i, 0, 0) = "PayLoad" Then
                bInstalled = - 1
        End If
        If MacroName$(i, 0, 0) = "FileSaveAs" Then
                bTooMuchTrouble = - 1
        End If
Next i
If Not bInstalled And Not bTooMuchTrouble Then
        'add FileSaveAs and copies of AutoOpen and FileSaveAs.
        'PayLoad is just for fun.
        iWW6IInstance = Val(GetDocumentVar$("WW6Infector"))
        sMe$ = FileName$()
        sMacro$ = sMe$ + ":Payload"
        MacroCopy sMacro$, "Global:PayLoad"
```

Jellegzetes részlet a hirhedt Concept vírusból, amely a globális sablonban való jelenlétet ellenőrzi és makrómásolást végez

```
For i = 1 To iMacroCount
        If WordBasic.[MacroName$](i, 0, 0) = "PayLoad" Then
                bInstalled = -1
        End If
        If WordBasic.[MacroName$](i, 0, 0) = "FileSaveAs" Then
                bTooMuchTrouble = -1
        End If
    Next i
    If Not bInstalled And Not bTooMuchTrouble Then
        'add FileSaveAs and copies of AutoOpen and FileSaveAs.
        'PayLoad is just for fun.
        iWW6IInstance=WordBasic.Val(WordBasic.[GetDocument
        Var$]("WW6Infector"))
        sMe$ = WordBasic.[FileName$]()
        sMacro$ = sMe$ + ":Payload"
        WordBasic.MacroCopy sMacro$, "Global:PayLoad"
```

Ugyanannak a Concept vírusnak Word97 alá konvertált, és ebben a környezetben WM97.Concept névre hallgató változata

ban már ebben a formában él tovább. Ez a jelenség teljesen új a vírusvilágban, még nem volt rá példa, hogy egy új platform megjelenésekor automatikusan áttelepíthetők lettek volna a vírusok. A vírusíróknak kisujjukat sem kellett mozdítaniuk ahhoz, hogy megfertőzzék az Office 97-et. Némi rosszmájúsággal az első kereskedelmi vírusgyártó csomagnak is lehetne titulálni az új Word verziót.

Erre illett volna odafigyelniük a Microsoft szakembereinek, hiszen mi sem lett volna egyszerűbb, mint a konverzió közben egy röpke víruselllenőrzés. Bizonyos jelek utalnak persze arra, hogy ez a Microsoft mérnökeinek fejében is felmerült. Ha ugyanis valaki figyelmesen végignézi a Word97 csomag állományait, akkor meglepve tapasztalhatja, hogy a WWINTL32.DLL tartalmazza a legelterjedtebb vírusok (Concept, NPAD, Bandung stb) szignatúráit.

A Microsoft saját bevallása szerint (a fontos technikai információkat tartalmazó Knowledge Base egy cikke alapján) e könyvtár többek között azt a célt szolgálná, hogy a konverzió folyamatában felismerje és eltávolítsa a legfontosabb makróvírusokat. Nos, amint az egyszerű kísérletezéssel eldönthető, egyáltalán nem akadályozza meg a fent említett vírusok konverzióját, valami okból úgy dönthettek, hogy eltávolítják ezt a funkciót.

Vagyis felmerült a konverzió alatti víruselllenőrzés kérdése, tettek is bizonyos lépéseket, de azután hamvába holt a terv. A végső verzióban ezeken a (véletlenül bennfelejtett) stringeken kívül semmi jele nincs a tudatos vírusvédelemnek. Az egyetlen eredmény az, hogy néhány víruskereső éppen e bentmaradt stringek miatt véli fertőzöttnek a szóban forgó fájlt, nem kis riadalmat okozva a felkészületlen felhasználóknak.

A natív vírusok

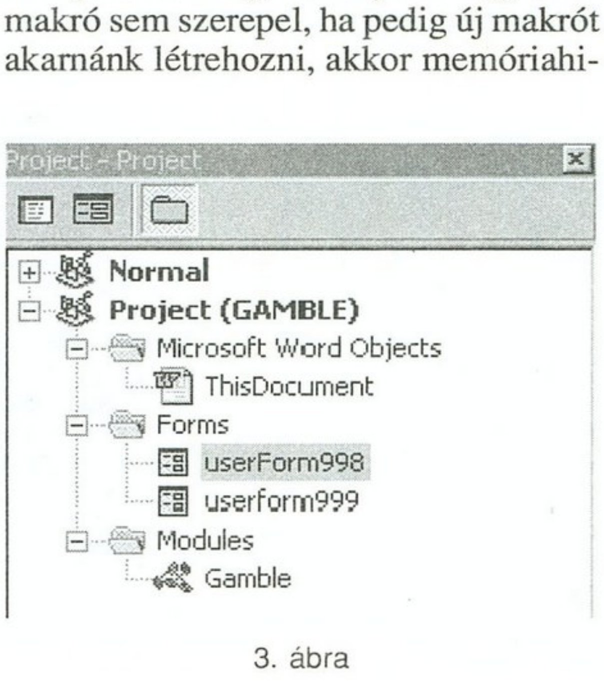
Persze a vírusírók nem olyan gyerekek, hogy mindent ráhagyjanak az automatikus konverzióra. Nagyon gyorsan alkalmazkodtak az új körülményekhez. A Word 6.0/7.0 esetében a vírusok leggyakrabban az alábbi két módszert használták az aktivizálódásra: az automatikus makrókat és a Word-parancsok átdefiniálását. Ha egy Word 6.0 dokumentumban a makró AutoOpen névre hallgatott, akkor az megnyitáskor magától lefutott. Ha volt egy FileOpen nevű makró, akkor a File menü Open pontjának kiválasztásakor az futott le a beépített Word parancs helyett.

A Word97 programozásának már nem a makrók, hanem a modulok az alapelemei. Minden modul egy önálló programozási egység, amelyen belül eljárások és függvények fordulhatnak elő. Ez annyiban változott a Word97 esetében, hogy egy ilyen automatikus makró kétféleképpen is megjelenhet: ha a dokumentum tartalmaz olyan modult, amelynek a neve AutoOpen, és van benne egy MAIN nevű eljárás, illetve ha valamely modulnak van egy AutoOpen nevű belső eljárása. Ilyenkor a dokumentum megnyitásakor ez az eljárás fut le. Ugyanez a helyzet a fent említett FileOpen makró esetében is. Az első megoldás (a MAIN eljáráson keresztüli aktivizálódás) az automatikusan konvertált vírusok esetében játszik szerepet, ennek makrói ugyanis pontosan olyan szerkezetű modulokká konvertálódnak, amelyek belépési pontja a MAIN nevű eljárás. A második eset, azaz a tetszőleges modulon belüli AutoOpen nevű eljárás esete már a VBA filozófiáját követi, és a már Word97-ben íródott, natív vírusok eszköztárába tartozik.

Az első fecskék

Az Office 97 boltokba kerülése után alig pár hetet kellett várni az első, már valóban VBA-ban megírt vírusra, a NightShade-re. Ez egy nem túl bonyolult vírus, amely terjedéséhez az AutoClose makrót használja fel. Ezenkívül két akcióval hívja fel magára a figyelmet: egyrészt átlagosan minden hetedik dokumentum bezárásakor a Word97 .NightShade by Pyro [VBB] üzenetet jelenteti meg az Assistant-ben, és péntek 13-án az éppen bezárt dokumentumokat a NightShade jelszóval titkosítja.

A másik vírus, a Gamble már fejlett lopakodó vírus: a Tools|Macro parancsot eltéríti, és olyan, a Concept-nél bemutatotthoz hasonló párbeszédablakot jelenít meg, amelyben egyetlen makró sem szerepel, ha pedig új makrót akarnánk létrehozni, akkor memóriahiányra hivatkozva megtagadja a műveletet. Valójában a vírus (amint azt a 3. ábra is mutatja) két párbeszédablakból és a kódmodulból áll.



3. ábra



4. ábra

A kódmodul egy AutoOpen és egy ToolsMacro eljárásból áll. Az utóbbi jeleníti meg a már említett szimulált párbeszédablakot, az előbbi pedig a vírus terjesztését végzi. Ezek alapján a Gamble a fertőzött dokumentumok megnyitásakor bemásolja magát a globális sablonba, ezek után pedig minden újonnan megnyitott dokumentumot megfertőz. Keddenként a vírus minden dokumentum megnyitásakor egy kis játékot játszik velünk. Először megjeleníti a mellékelt ábrát (4. ábra), majd pedig bekér egy 1 és 10 közötti számot. Ha ez a szám 5, akkor a vírus kitörli magát a globális sablonból és az aktív dokumentumból, ezzel ideiglenesen megszabadultunk tőle. Ha bármilyen más számot adunk meg, akkor egyrészt kitörli magát, másrészt az aktív dokumentumot a Gambler jelszóval védi le.

A Gambler után egyre-másra jelentek meg az újabb vírusok. A cikk megírásakor mintegy 50 Word97 alatti makróvírus létezett. Ezek kb. kétharmada konvertált, egyharmada pedig natív. Várható, hogy a nyári uborkaszezon után (amely érdekes módon egybeesik az egyetemi szünidővel is) a Word 6.0/7.0 vírusokhoz hasonlóan a Word97-vírusok száma is robbanásszerűen meg fog nőni. Szerencsére ez már korántsem éri annyira felkészületlenül a víruskeresők íróit, mint amekkora meglepetést a Conceptnek és társainak felbukkanása okozott. A harc pedig töretlenül folyik az írók és az irtók között, csak eggyel több harctéren.

Szappanos Gábor

Visual DataFlex 4.0

Munkaeszköz igazi profiknak

A DataFlex PC-s pályafutása a nyolcvanas évek közepén kezdődött. Az addig csak unixos környezetre fejlesztett programot a PC-piac robbanásszerű növekedése láttán adaptálták a személyi számítógépek világára. Már akkor is nagy ázsiója volt a DataFlexnek a programfejlesztők körében. Nem „Macilaciknak" készült, ezért széles körben nem terjedt el. E pozícióját a mai napig is megtartotta, viszonylag szűk, de annál felkészültebb tábor használja. Vonzó tulajdonsága, hogy adatbázisa, programkódja (forrásszinten) több operációs rendszerben is azonnal megél. Most a Visual DataFlex 4.0-s változatáról osztom meg első benyomásaimat olvasóinkkal.

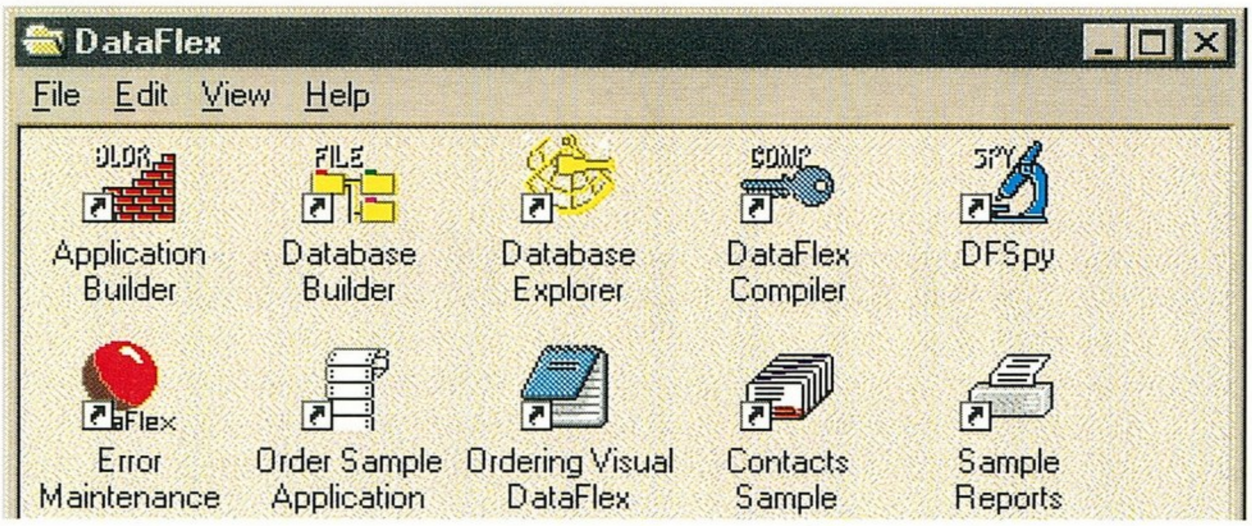
A Visual DataFlexet előszeretettel használják nagyobb adatbázisok kezelésére. Rendkívüli képessége, hogy az index alapján történő rekordelérési idő többmilliós rekordszám esetén is csak néhány tized másodperc. Mára 32 bites lett, Windows 95 és NT 4.0 alatt hajlandó futni. Klasszikus tulajdonságait elődeihez méltóan megtartotta, továbbra is a professzionális fejlesztőeszközök közé tartozik. Windows 95 és NT rendszerű hálózatokon kívül a Novell 3-as és 4-es szériáját támogatja. Más hálózatokkal nem törődik, de ez talán nem is olyan nagy tragédia.

Különleges adatbázis

Ejtsünk néhány szót elöljáróban az adatbázis tulajdonságairól és teljesítőképességéről. A DataFlex ISAM struktúrájú relációs adatbázissal operál. Tetszőleges számú adatbázis-terület definiálható, területenként 250 fájllal. Egy fájl maximális rekordszámának gyakorlatilag csak a merevlemez kapacitása szab határt (16,7 millió). Egy rekord maximum 16 Kbájt méretű lehet. Az adatfájl változó rekordhosszúságú, a Text és Bináris mezők rekordonként tetszőleges hosszúak lehetnek.

A numerikus adatok BCD formátumban, a dátum típusú mezők három bájton numerikusan, míg a sztringek háromféle tömörítési eljárás valamelyikével tárolódnak: standard, fast, custom (karaktergyakoriság-függő). Érdekesség az Overlap mezőtípus, amely logikai hivatkozást teremt egy tetszőleges méretű rekordrészletre. Így egy paranccsal több mező is átmásolható, törölhető stb. Egy fájlhoz 16 index rendelhető. Egy index 16 mezőből állhat, hossza maximum 255 bájt lehet. Az indexek két fajtáját különbözteti meg, az online és az offline indexet. Az online index a rekordműveletek alkalmával azonnal aktualizálódik, az offline csak az újraindexelés alkalmával. Nagyobb adatbázisoknál, tömeges adatbevitel esetén előszeretettel használják ez utóbbi verziót. Az állandó fejlécintegritás-védelem figyelmeztet az adatok sérülésére, és megakadályozza a további műveletvégzést. Háromféle tranzakcióvédelem közül választhatunk, ezek a „client atomic", „server atomic" és „server logged" módok.

E nevek a normál védelmet, a munkaállomásról kezdeményezett tranzakció totális védelmét és a hálózatos, objektumorientált szerver-kliens tranzakciókezelést takarják. Itt állítható be a lockolás fájl- vagy rekordszintre. Rendkívül fontos megemlíteni a DSO (Data Server Object) szerepét. Ez a programozó által definiált logikai szerver biztosítja a tranzakciók alatt az adatbiztonságot, a kitörölt, megváltozott adatok eljutását az éppen inaktív objektumokhoz, ellátja a szükséges



egyéb szinkronizálási feladatokat. E feladatokat már az adatbázis tervezésekor pontosan meg kell határozni.

Lényeges újdonságok

A DataFlex nem biztosít online kapcsolatot a populáris adatformátumokkal. Alapfilozófiája következtében ebben a rendszerben mindent meg lehet oldani, ezért csak végszükség esetére tartogatja a dBase-fájlt importáló és a szöveges fájlt exportáló rutint.

A Visual DataFlex fejlesztői környezet mindössze néhány modulból áll, ezek viszont nagyon hatékonyak. Az egyszerű elérhetőség kedvéért a desktopon megjelenik a DataFlex powerbar a legfontosabb modulok ikonjaival. A hagyományos eszközök is megmaradtak, ezekről is ejtünk néhány szót.

Az előző verzióhoz képest újdonság a Workspace, Data dictionary fogalma, a Database Builder és az Application Builder. Alapvető koncepcióját User Interface Management Systemnek nevezik, az ott megfogalmazott alapelvek a felhasználói felület logikai felépítésére, a beszámolók menetére, a helyzetérzékeny help és a hibakezelés kivitelezésére vonatkoznak. Érdemes ezen alapelvek szerint eljárni, és akkor garantált a gyors, hatékony programfejlesztés.

Workspace

Más fejlesztőeszközök projekt névvel illetnék, de ez egy kicsit más. Nemcsak a programok halmazát rögzíti egy kupacba, hanem a hozzájuk tartozó minden egyéb csecsebecsét, mint a data, report, screen, help, image, def fájlokat is. Így ugyanazon fejlesztőeszköz munkaterületenként más-más tulajdonságokkal rendelkezhet, a máshol használt anyagok nem zavarják a munkát. Természetesen megvan a lehetőség arra is, hogy a munkaterületeket bármilyen szinten összekapcsoljuk.

Data Dictionary

A munkaterület fontos része ez a könyvtár. Itt tároljuk az adatbázisra vonatkozó jellemzőket, fájlneveket, mezőtípusokat, hosszokat, relációkat, az adatmezők értékvizsgálati, adatmentési jellemzőit, adatintegritási ellenőrzéseit és egyéb eljárásokat. Fontos része lesz a programkészítésnek, mert az itt definiált tulajdonságok a programgeneráláskor majd automatikusan érvényesülnek.

Database Builder

A rendszer legfontosabb eszközeinek egyike. A programkészítés előtt itt definiáljuk az adatfájlok felépítését, az indexeket, a relációkat. A manapság oly divatos wizardok (varázslók) segítségével definiálható új munkaterület, vagy

módosítható egy már meglévő. A szokványos funkciók mellett itt állítható be a tranzakcióvédelem és a lockolás módozata. A DataFlex precíz rendszer, azonnal elkészíti az adatfájl dokumentumszerű adatlapját, a hozzá tartozó böngészőprogramot, és elhelyezi azokat az adatterület könyvtárában.

Database Explorer

Jólesõ érzés az általunk kreált adatbázist „madártávlatból" megtekinteni. Itt könnyedén ellenőrizhető a létrehozott struktúra helyessége, a fájlok adattartalma. Jó áttekinthetősége komoly hibakeresővé avatja ezt az első pillantásra egyszerű eszközt. Kicsi a bors, de erős! Közvetlenül módosítható bármelyik fájl tartalma, ezért használata fokozott körültekintést igényel, hiszen itt minden ellenőrzés nélkül bűvészkedhetünk az adatokkal.

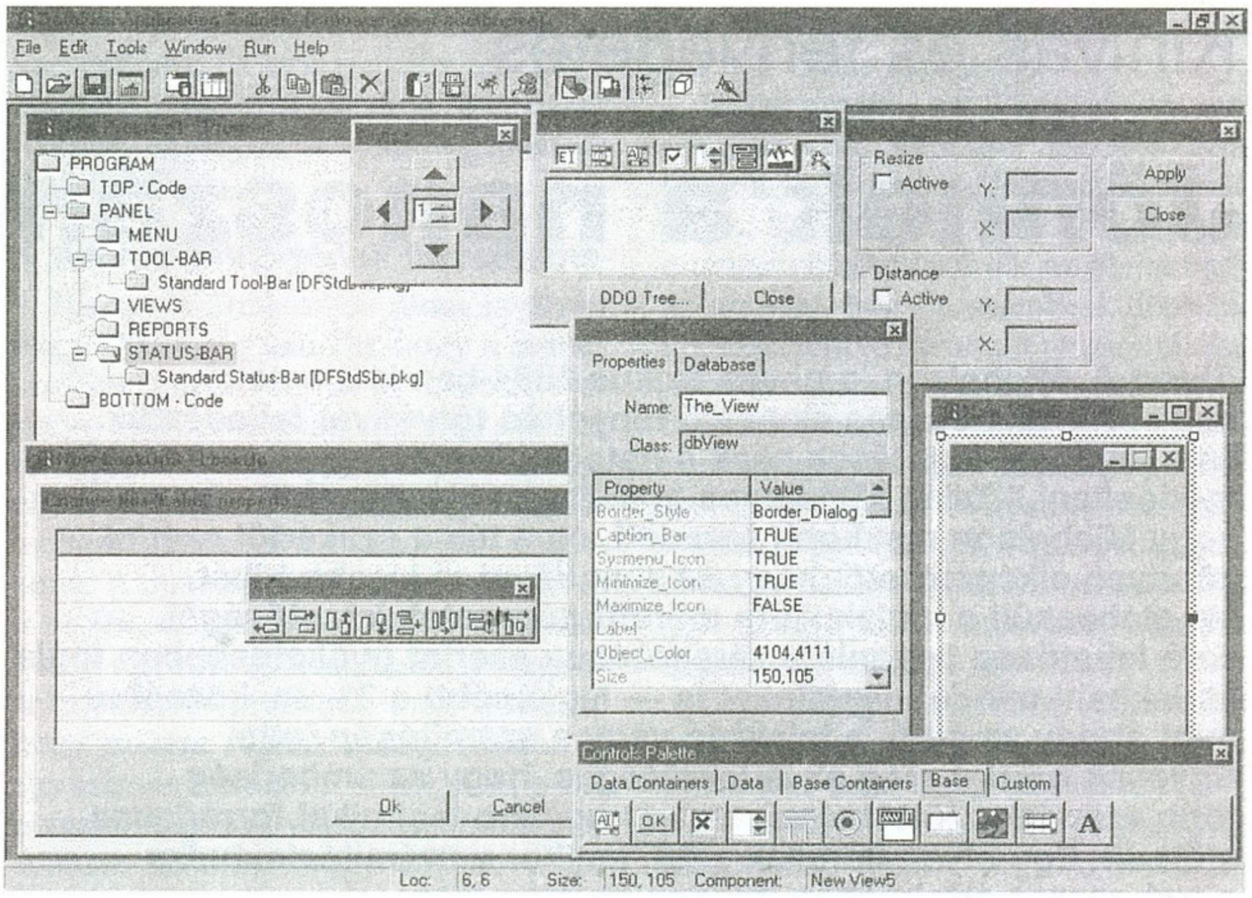
Application Builder

A programkészítés színtere. A Database Buildernél megismert módon Workspace felosztásban különítjük el programjainkat, adatainkat. Első lépcsőben célszerű egy rövid tesztprogramot készíteni, mely az adatbázis összes elemét, relációját, a Data Dictionaryben definiált összes procedúrát felhasználja. Így gyorsan kiszűrhetők az esetleges definíciós hibák. Ha ezen túljutottunk, kezdődhet az érdemi programgenerálás.

Miként az egy jól szituált objektumorientált fejlesztőrendszerhez illik, a dialog editorral gyorsan haladhatunk előre az ablakok gyártásával. Az előre elgondolt programkép hamar elkészül, az adatmezők és ellenőrző procedúrák egymáshoz rendelése egy kicsit több időt vesz igénybe. Itt térül meg az, ha a legelején az adatbázis tervezésével egy kicsit többet foglalkoztunk.

Természetesen sokkal többről van szó, mint egyszerű összerendelésekről, mert azonnal rendelkezésre áll a működőképes forráskód is, amelyet saját igényeink szerint bővíthetünk tovább. Kukkantsunk bele egy kicsit a forráskódba! Pascalhoz hasonlatos, mégis egy parancsnyelvet látunk. Nagyon szimpatikus, hogy az automatikusan generált kód a megértéshez elegendő megjegyzést tartalmaz, így a nyelv ismeretében könnyű a továbbfejlesztés. Bár idáig bárki könnyedén eljuthat, a DataFlex kellő ismerete nélkül nem ajánlatos tovább merészkedni, mert innen már meredekebb az út.

Ha már tökéletesen működik programunk, nekiláthatunk a jelentések (riportok) elkészítéséhez. Kétféle út lehetséges, a hagyományos és a WinQL runtime változatával elérhető azonnali jelentésgenerálás. Lehetőség van külső objektumok beillesztésére is, a WinQL által a Borland C++, a Crystal Report Print Engine és Translation Library is a rendszer része.



DFSpy

A végső hibakeresés eszköze, amellyel minden rejtett hiba felderíthető. Tetszőleges számú feltételes vagy feltétel nélküli töréspont helyezhető el, illetve egyéb, feltételhez kötött töréspontok is felállíthatók. A hibakeresés közben minden objektum állapotáról, a különböző pufferek tartalmáról informálódhatunk, sőt azokat el is menthetjük további elemzéshez.

Helyzetérzékeny helprendszer

Tekitve, hogy a súgót a Microsoft használható módon megoldotta, a Data Access ezzel nem bajlódott, hanem beépítette rendszerébe, így szabványos MS helprendszert kell készíteni az applikációkhoz. Segítségül Hotspot editort és egy bitmap-konverterprogramot mellékeltek, így a grafikus részek jól beilleszthetők.

A fejlesztői CD-n néhány mintaalkalmazás is található. Érdemes végignézni belső felépítésüket és működésüket, mert rávezetnek arra az útra, amelyen DataFlex programokat érdemes fejleszteni. Néhány alkalmazással a teljes palettát bemutatják, bár mindenkinek lehetnek újabb ötletei. Így például, aki belepillant az FMAC fájlba (ez a DataFlex utasítástára), láthatja, hogy a megszerzett tudással újabb parancsok készíthetők egyedi igényeinknek megfelelően. Ezt érdemes is kihasználni, sok többletmunkától kímélheti meg magát a vállalkozó szellemű programozó. Az is kiderül, hogy bár teljesen objektumorientált a rendszer, továbbra is lehetséges a hagyományos procedurális programozás. Néhány helyzetben erre szükség lehet, és jó tudni, hogy ez az eszköz alkalmas rá.

Értékarányos ár

A Visual DataFlex olyan professzionális adatbáziskezelő, melynek ára társainak átlagosan két-háromszorosa, teljesítményben, az általa nyújtott adatbiztonságban is messze felülmúlja azokat. Nem mindenhol a legelegánsabb a képernyőkezelése, nincs annyi kényelmi szolgáltatása, csillogása, varázslása, de az alapos szemlélő észreveszi, hogy itt nem a külcsín, hanem a belbecs, a tartalom a lényeg. Komoly, biztonságos rendszerek készíthetők vele, s borsosnak tűnő ára nem is oly sok, ha figyelembe vesszük, hogy a vele készült programok milyen jól forgalmazhatók. Felkészült programozó kezében már az első feladat megoldása megtéríti az árát.

A DataFlex munkahely javasolt minimális konfigurációja: Pentium processzor, Windows 95 vagy NT 4.0, és a két operációs rendszertől függően 16 vagy 32 MB RAM. A merevlemezen 80 MB szabad hely kell, ebből az üres rendszer kb. 50 MB-ot foglal el a mintapéldákkal együtt. A dobozban megtalálható könyvek Word 6.0 formátumban, az egyéb online referenciák MS Help formájában találhatók a CD-n.

Azon programfejlesztők figyelmébe ajánlható e program, akik szakmai fejlődésük érdekében hajlandók a szokásostól kissé eltérő rendszert is megismerni és alkalmazni.

Veress Gábor

Kihívás és fenyegetés

Jönnek a nanoszámítógépek?

Albert A. Michelson, a neves fizikus 1894-ben kijelentette, hogy „A fizika összes alapvető tényét és törvényét felfedeztük... és a jövőbeni kutatások csak a hatodik tizedesjegyértéket fogják módosítani." Talán fölösleges hosszasan bizonygatni, hogy Michelson mekkorát tévedett, és a téma szakértői szerint könnyen elképzelhető, hogy az sem követ el kisebb hibát, aki alábecsüli a molekuláris méretű gépezetek jelentőségét. Ez a lehetőség ugyanis a várakozások szerint gyökeresebben fogja átalakítani mindennapjainkat is — legkésőbb a 21. sz. közepére —, mint ahogy az autó, a televízió vagy a számítógép tette. Egyesek szerint még az is lehetséges, hogy az emberiség több ezer éves történelmének legnagyobb technikai forradalma előtt áll. Egy olyan korszak előtt, amikor a számítástechnika a szó szoros értelmében mindenütt jelen lesz.

Richard Feyman, a később Nobel-díjat kapott fizikus már 1959-ben kijelentette, hogy meggyőződése szerint „a fizika törvényei nem zárják ki annak lehetőségét, hogy atomonként mozgassuk a dolgokat", és mai tudósok úgy vélik, hogy tökéletesen igaza volt: szabad az út a feyman-i elképzelés megvalósítására. És a nanotechnológia az a tudomány, amely az ebből következő, esetleg valóban felmérhetetlen horderejű lehetőségekkel foglalkozik.

A molekuláris mindenhatóság

Bizonyos szempontból nincs nagy különbség a pattintott kőeszköz és a 600 MHz-es, 64 bites szilikonchip között: mind a kettő arra szolgál, hogy az ember ilyen vagy olyan képességeit fokozza, és — ami a számunkra most fontosabb — mind a kettő ugyanazzal a technológiával készül. Akár az obszidiánpenge, akár a mikroprocesszor esetében az anyag megszámlálhatatlanul sok atomot tartalmazó, nagyobb darabjait munkáljuk meg, és közben nem törődünk vele, hogy elvileg manipulálhatnánk a kisebb méretek szintjein is, hogy a kívánt célt elérjük, és létrehozzunk egy meghatározott tárgyat. Ha megfelelő eszközökkel rendelkeznénk, akkor semmi sem indokolná, hogy a dolgoknak csak a felszínén maradjunk, és ne foglalkozzunk ugyanúgy a mikroszintű részletekkel a gyártási folyamatok során, mint ahogy a genetikus sem a tigris testét tanulmányozza, hanem a tigris testének felépítését meghatározó géneket. A különbség nagyjából akkora a hagyományos, „durva módszerek" meg a molekuláris, sőt, atomi szinten dolgozó technológia között, mint amekkora a hagyományos sebészet és a génsebészet közötti eltérés.

Nem nehéz elképzelni, mennyivel jobb minőségű, mennyivel szilárdabb, rugalmasabb, könnyebb és ellenállóbb fémeket lehetne előállítani, ha nem bíznánk a véletlenre semmit, hanem molekuláról molekulára, netalán atomról atomra haladva határoznánk meg az ötvözetek összetételét. Végső soron akár tökéletes és hibátlan alapanyagokat is lehetne gyártani, és ez már önmagában is megérne minden fáradozást: óriási jelentősége volna a repülőgépgyártástól a számítógépiparig bezárólag mindenütt.

Ez azonban legfeljebb az első lépés, és a hosszú út későbbi szakaszain valahol már csak afféle molekuláris mindenhatóság volna, hiszen bizonyos szempontból egy növény vagy egy állat sem más, mint hidrogén-, oxigén-, nitrogén-, szén- és néhány másfajta atom megfelelő elrendezése. Tehát legalábbis elvileg nem lehet semmi akadálya annak, hogy megfelelő ismeretanyag és megfelelő technológia birtokában atomról atomra haladva bármit megépítsünk, ami egyáltalán elképzelhető. Lehetséges eredményeit, illetve a lehetséges visszaéléseket tekintve ehhez képest a génmérnökség talán nem is olyan nagyon fenyegető dolog.

Milliárdod léptékű tudomány

Mivel a „nano" előtag a mögötte álló kifejezés ezermilliomod részét jelenti, ebből könnyen kitalálható, hogy a nanotechnológia egy, a hatását a milliméter milliomodrészének nagyságrendjében kifejtő technológia.

K. Eric Drexler, aki 1986-ban először foglalkozott ezzel a hipotetikus, új tudománnyal, abból indult ki, hogy létezik a molekulákat „pozicionálni" képes gépezet a természetben is: a biológiából ismert riboszóma ugyanis molekuláról molekulára építi fel egy ember, egy állat vagy egy növény testét alkotó fehérjéket, miközben az ún. mRNS-től (messenger, azaz hírvivő RNS) kapja az arra vonatkozó utasításokat, hogy mit hová illesszen. És ha most úgy képzeljük el a riboszómát, mint a milliméter milliomodrészének léptékében tevékenykedő robotgépet, az mRNS-t pedig mint a robotgépet vezérlő programot, akkor máris nyilvánvalónak tűnik a megoldás: először készíteni kell egy ilyen nanomechanizmust, a nanotechnológiában bevett szóhasználattal élve egy assemblert, majd pedig egy megfelelő, molekuláris méretű komputert is, ami ezt a műriboszómát vezérelni tudja. A számítások azt mutatják, hogy bár egy ilyen miniatűr számítógép (megfelelő háttértárlooval és 1 KB RAM-mal) nagyobb lenne a riboszómánál, a mérete még mindig bőven alatta maradna egy baktériuménak.

Az így létrejövő nanogépezetek azután a fentebb már említett, minimális célkitűzéseknek megfelelően — és a riboszómákkal ellentétben — képesek lehetnek nem csupán romlékony fehérjeláncokat, de bármilyen más anyagot is előállítani, és ennek ismeretében már nem is olyan túlzó Drexler azon kijelentése, mely szerint „A számítástechnika, az orvostudomány, az űrkutatás technikái és termékei — és persze a háború is — az atomok átrendezésére irányuló képességeinken múlik. Az assemblerek segítségével pedig újraépíthetjük vagy éppen lerombolhatjuk a világot."

De a dolog azért nem olyan egyszerű. Például okáért, még ha meglenne is a létrehozásukhoz szükséges, teljesen pontos elmélet, nem kezdhetünk hozzá rögtön a milliomodmilliméter méretű gépezetek építéséhez: előbb nagyobb szerkezeteket kell létrehozni, amelyek

segítségével elkészíthetők a kisebbek, hogy majd segítségükkel még kisebbeket csináljunk stb., és a mai technológia erre még nem nagyon alkalmas. Közkeletű hasonlattal élve jelenleg mintha bokszkesztyűvel próbálnánk pozicionálni egy homokszemet, úgyhogy ennek megfelelően lehetőség szerint óvatosan kell megítélni a mégoly ígéretes eredményeket is.

Például azt, amit az IBM kutatólaboratóriumában értek el, amikor sikerült egy ún. pásztázó alagútmikroszkóppal, elektronnyalábok segítségével mozgatva atomokból kirakni az IBM betűit: ezzel az atomi szintű tevékenység gyakorlati lehetségességét bizonyították ugyan, de a módszer egyelőre távol van a használhatóságtól. És könnyen lehet, hogy nem ez lesz a jövő útja, noha egyes elképzelések szerint továbbfejlesztésével esetleg akár új anyagokat, rendkívül kedvező tulajdonságú vegyületeket, meg kisebb, olcsóbb áramköröket is elő lehet majd állítani.

De legalább ugyanennyi, ha ugyan nem több reménnyel kecsegtet a kémia egy egészen speciális, lendületesen fejlődő ága, az ún. positional chemistry: ma már egyáltalán nem számít lehetetlennek bizonyos vegyületek tökéletesen pontos, atomról atomra történő szintetizálása (sőt, olykor az eredetinél ellenállóbb és „jobb" fehérjemolekulák előállítása sem, miként az a Du Pont kutatóinak sikerült még a '80-as évek végén), és bár a végeredményként létrejövő struktúrák rendszerint meglehetősen kisméretűek, elvileg nincs semmi akadálya, hogy a pár száz atomból álló molekulákból bonyolult, háromdimenziós szerkezeteket rakjunk össze.

Ezek azután például egy számítógép logikai egységeiként szolgálnának, mintha csak a Tinkertoy nevű (pálcikákból és korongokból álló) játék darabjai lennének. De amíg a makroelemekből, „durva technológiával" készült mechanikus komputerek elviselhetetlenül lassúak, addig jó esetben akár 1/100-ad köbmikronba beleférne egy egyszerű nanoszámítógép, és ha az egész, több milliárd elemből összetevődő készülék olyan „nagy" volna is, mint egy baktérium, még így is megfelelően gyors maradna, mert a mechanikus alkatrészeknek jelen esetben igen kis távolságot kell áthidalniuk.

Ha valaki azt kérdezné, hogy miért éppen nanokomputert akarnánk építeni, akkor a válasz az lenne, hogy azért, mert a nanotechnológia tulajdonképpen azt a célt tűzte ki maga elé, hogy az elképzelhető legkisebb méretek szintjén is kontrollálja az eseményeket, és ehhez nyilvánvalóan számítógépekre van szükség. És az is nyilvánvaló, hogy ebből a megközelítésből a fehérjéket felépítő egységek is számítógépeknek tekinthetők.

A nanoszámítógépek forradalma

Bizonyos értelemben nincs is olyan nagy különbség aközött, hogy a természetben „készen található" assemblereket — a riboszómákat, valamint a létező fehérjéket — manipulálva próbáljuk-e elérni céljainkat, vagy pedig a nanotechnológia „saját eszközeivel" dolgozunk. Annál nagyobb viszont az eltérés a kétféle szemlélet között. A biokémikusok ugyanis tipikusan tudósi megközelítéssel azt próbálják meg kitalálni, hogy miként fognak elhelyezkedni és egymáshoz kötődni a molekulák, a nanotechnológusok viszont mérnöki feladatnak tekintik (vagy inkább majd fogják tekinteni) a problémát, és megpróbálják megtervezni, hogy miként illeszkedjenek egymáshoz az atomok.

Drexler forgatókönyve szerint ennek az utóbbi szemléletmódnak az elterjedése, meg persze a természettől készen kapott proteingépezetek széles körű és céltudatos használata fogja eredményezni az új technológia első korszakának beköszöntét, de ennek a módszernek még meglesznek a maga korlátjai. „Szerszámai", a fehérjék például csak meglehetősen limitált hőmérsékleti tartományon belül működőképesek (túl hidegben megfagynak, túl melegben megfőnek). Ahhoz azonban még mindig elég jónak bizonyulnak majd, hogy segítsék létrehozni az első igazi nanogépezeteket, amelyek azután viharos gyorsasággal fognak mindenütt elterjedni.

Már csak azért is, mert a teória szerint az élő anyaghoz hasonlóan önreprodukciós képességekkel fognak rendelkezni (aminek elvileg Neumann János 1940-es tanulmánya óta nincs semmiféle akadálya), és emellett természetesen képesek lesznek arra is, hogy önmaguk reprodukálásán kívül az emberiség számára fontosabb más tevékenységet is folytassanak (máskülönben teljesen feleslegesek volnának). Ennek eredményeként a számítástechnika hamarosan a szó szoros értelmében mindenhova beszivárog, ezek a nanomasinák: ezek a többé-kevésbé univerzális, miniatűr számítógépek ugyanis programjuknak megfelelően minden elképzelhetőt, vagy majdnem mindent képesek lesznek megvalósítani. Amennyiben erre sor kerül, akkor elmondhatjuk, hogy a számítástechnika ténylegesen és alapvetően megváltoztatja az életünket (még ha nem is pontosan úgy, ahogyan azt Bill Gates elképzeli).

Ekkor ugyanis a különböző nanomechanizmusok képesek lesznek arra, hogy a hulladékból élelmiszert csináljanak, hogy megtisztítsák a levegőt, és a tengervízből kiszűrjék az olajat vagy az aranyat, hogy a szembe különböző képeket vetítve háromdimenziós tájakat jelenítsenek meg, és hogy hosszú távon beférkőzve az emberi szervezetbe előbb mesterséges immunrendszerként védelmet biztosítsanak a vírusok ellen, majd a rákot számolják fel, végül pedig gondoskodjanak az egyes ember lehetőségek szerinti minél hosszabb, már-már örökkévaló életéről is.

És amennyiben valaki kételkedne mindezek lehetőségében, akkor bátran gondolhat arra, hogy (legalábbis elvileg) a fent említett minden probléma megoldható atomi vagy molekuláris szintű manipulációkkal, ha pontosan tudjuk, hogy azt miként kell csinálni. Hiszen a szemét és a fehérjedús, tápláló étel egyaránt atomokból áll, a rák pedig olyan, kórosan burjánzó sejtek tömege, amelyeket egyszerűen el kell távolítani vagy „felül kell írni", azaz át kell alakítani. És az öregséggel is az a helyzet, hogy ha minden sejt vagy sejtcsoport mellé afféle megfelelően programozott „nanorendőrt" állítunk, amely szükség esetén aktívan beleavatkozik az anyagcsere-forgalomba, akkor az anyag soha nem fárad el.

Aktív pajzsok és nanoterroristák

Vagyis a nanotechnológia mintha a megvalósult utópiát ígérné, és mint ilyen, végtelenül csábítónak tűnik. És tegyük hozzá rögtön, hogy végtelenül veszélyesnek is. Vannak egyáltalán nem molekuláris méretű árnyoldalak...

Nem kétséges ugyanis, hogy még amennyiben valóban meghatározó jelentőségűvé válna is a nanotechnológia, a dolognak lennének szándékos és szándékolatlan hátrányos hatásai. Ami az előbbieket illeti, nem kell hozzá különösebben élénk fantázia, hogy a földi paradicsom helyett elképzeljük, amint valaki nem az agysejtek regenerációjára és a gondolkodási képesség hosszú távú megőrzésére használja fel a nanotechnológia eredményeit, hanem a feltehetően egyszerűbben megvalósítható nano-kábítószerezésre: mindössze úgy kell programozni ezeket a molekuláris gépezeteket, hogy állandóan stimulálják az agyi gyönyörközpontot, és már készen is van egy minden eddiginél hatásosabb drog. Amit azután persze a katonaság is felhasználhat a saját céljaira: nem véletlen, hogy a hadiipar már

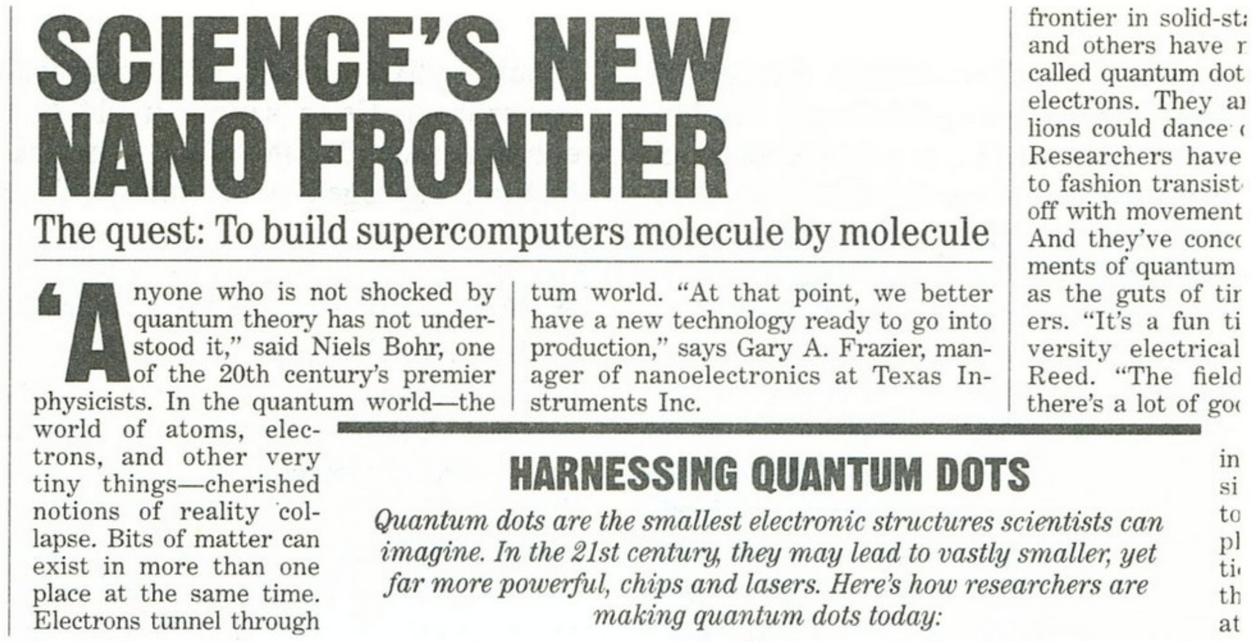
most is komolyan érdeklődik a nanotechnológia iránt.

De kidolgozhatnak lényegesen hatékonyabb vagy riasztóbb módszereket is az ellenség legyőzésére: a műholdról pozicionált csapásmérő nanogépezetek például pontosan olyan hatékonyak lesznek, amilyenre programozzák őket, és a vegyi fegyverek tömegpusztító hatása kombinálódik bennük a távirányítású rakéta pontosságával. Megfelelő beállítások révén egész országok teljes népessége lesz kiirtható, miközben a határ túloldalán, pár méterrel odébb senkinek nem görbül meg a haja szála sem, és legfeljebb az az egyelőre legfeljebb nagyon is hipotetikus formában létező AS, vagyis Active Shield (aktív pajzs) fog védelmet kínálni ellenük, ami tulajdonképpen ismét csak mérnökileg megtervezett, mesterséges immunrendszer volna. Ezúttal azonban nem az influenzavírusok ellen.

Az erre a célre létrehozott nanorobotok nevükkel összhangban az ellenséges szerkezeteket felderíteni igyekezve, afféle aktív pajzsként védekeznének a behatolók ellen: valahogy úgy képzelhetjük el a dolgot, mint a számítógépes vírusok és a vírusirtók közötti versenyfutást, azzal a különbséggel, hogy ez esetben nem egy harddiszk épsége, hanem a mesterséges immunrendszerrel felszerelt ember élete a tét.

Hasonlóképpen felbukkanhatnak a számítógépes terroristák, a vírusírók mintájára a nanoterroristák is (egyáltalán nem véletlen, hogy katonai kutatóintézetek is aktívan foglalkoznak vírusfejlesztéssel), és ez még nem is a lehetséges legnagyobb rossz. Közben ugyanis arról sem szabad elfelejtkeznünk, hogy mint minden nagy hatású technológia esetében, itt is bekövetkezhetnek véletlen katasztrófák. A nanogépezetekkel kapcsolatban talán legtöbbször véletlen elszabadulásukat szokták emlegetni, de könnyen lehetséges, hogy ez nem is olyan fenyegető, mint első hallásra gondolnánk. Ahhoz ugyanis, hogy vírusként terjedve „felfalják" az egész Földet, az kellene, hogy a tervezettnél sokkal jobban hasonlítsanak az élő szervezetekre, a valóságban azonban közelről sem lesznek olyan flexibilisek, mint a mikroorganizmusok. A mostani elképzelések szerint inkább benzinüzemű autókra hasonlítanak majd, nem pedig mindenevő sertésekre, tehát a jelek szerint kicsi a valószínűsége, hogy „letérnek az útról": egyszerűen nem lesznek hozzá eléggé rugalmasak.

Más persze a helyzet, ha olyan nanobottal (nanobot=nanoméretű robot) van dolgunk, amely bármilyen alapanyagot felhasználva képes az önreprodukcióra. Az ilyen, gray-goo-nak nevezett szerkezeteket nanorendőrség gyanánt működve az ún. blue-goo-k lennének hivatottak kordában tartani, meggátolandó, hogy azok elszabadulva az egész bioszférát „robot-goo"-vá alakítsák, és végképp lehetetlenné tegyék az életet. Eközben a blue-goo-kat még arra is fel lehetne használni, hogy a robbanásszerű elszaporodás megakadályozása mellett törekedjenek a bioszféra rendben tartására: próbálják csökkenteni a légkör széndioxidtartalmát, hatástalanítani a veszélyes hulladékokat, és így tovább. Amivel esetleg csak rontanak a helyzeten.


SCIENCE'S NEW NANO FRONTIER

The quest: To build supercomputers molecule by molecule

"Anyone who is not shocked by quantum theory has not understood it," said Niels Bohr, one of the 20th century's premier physicists. In the quantum world—the world of atoms, electrons, and other very tiny things—cherished notions of reality collapse. Bits of matter can exist in more than one place at the same time. Electrons tunnel through

tum world. "At that point, we better have a new technology ready to go into production," says Gary A. Frazier, manager of nanoelectronics at Texas Instruments Inc.

frontier in solid-st and others have r called quantum dot electrons. They a lions could dance Researchers have to fashion transist off with movement And they've conc ments of quantum as the guts of tir ers. "It's a fun ti versity electrical Reed. "The field there's a lot of go

HARNESSING QUANTUM DOTS

Quantum dots are the smallest electronic structures scientists can imagine. In the 21st century, they may lead to vastly smaller, yet far more powerful, chips and lasers. Here's how researchers are making quantum dots today:

Az egyik igazi veszélyforrás ugyanis éppen az, hogy egy rosszul programozott blue-goo nem pontosan azt csinálja, amire szánták. A számítástechnika eddigi története fényesen bizonyította, hogy hibátlan program nincsen, és jól működő is csak olykor-olykor akad. Ez éppen elég gondot okozott az eddigiekben is, de a hatás egyenesen katasztrofális lehet, amikor a program nem egy számítógépen fejti ki a hatását, hanem közvetlenül a természeten. Itt ráadásul nagyon nyilvánvalón tetten érhető lesz a protein-assembler, illetve az élő szervezet-nanogépezet egymás mellé állításának korlátozott jogosultsága is: míg az előbbi esetben a nem rátermettek kihullanak a természetes szelekció rostáján, az utóbbiban az a blue-goo nem számít „életre alkalmatlannak", ami felmérhetetlen károkat okoz csak azért, mert egy programozó elnézett valamit.

Mégis nanotechnológia?

Így hát ajánlatos vigyázni a nanogépezetekkel, és arról sem szabad megfeledkezni, hogy ha a lehetséges mindenhatósággal élve durván és közvetlenül belenyúlunk például a természeti körfolyamatokba, akkor egyszerűen megjósolhatatlan, hogy mi lesz a következménye egy-egy lépésünknek. Vagyis még egy tökéletesen programozott, ám nem izolált körülmények között létező nanogépezet is irtózatos veszélyt jelenthet: egyszerűen azért, mert adott esetben képtelenek vagyunk előre látni a program tökéletes végrehajtásának minden áttételét és következményét.

Ám ebben nincs is semmi meglepő, és inkább azt kellene furcsállnunk, ha egy előreláthatóan olyan forradalmi jelentőségű technológiai váltás, mint amilyen a nanotechnológia is, csak pozitív lehetőségeket hordozna magában. Emellett nem szabad azt feltételezni, hogy egy eszköz vagy egy tudomány önmagában „jó" vagy „rossz", hiszen minden attól függ, hogy mire és hogyan próbáljuk felhasználni. Ráadásul arról sem kell elfelejtkezni, hogy még az előre látható technikai forradalmak sem pontosan olyan formában és nem annyira intenzíven köszöntenek be, mint ahogyan azt a teoretikusok várni szokták. Az 1780-as években, a léghajózás megjelenésekor azt hitték a szakemberek, hogy néhány éven, legfeljebb egy évtizeden belül mindenki repülni fog, és a földi közlekedés elsorvad, de ez természetesen nem következett be. És azoknak sem volt igazuk, akik azt jósolgatták, hogy a számítástechnika már a 21. század beköszöntére meghódítja a világot, és lépni sem fogunk tudni nélküle. Valójában azok az autók is tökéletesen alkalmasak a közlekedésre, amelyekben nincs beépített fedélzeti számítógép.

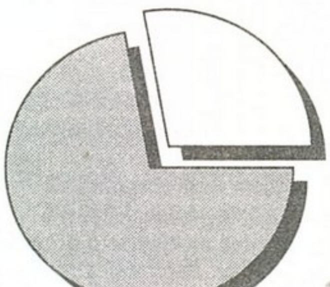
Hiba lenne azt remélnünk, hogy a nanotechnológia minden problémát egy csapásra megold, vagy legalábbis azonnal és alapvetően átformálja mindennapjainkat. Egyedül az látszik bizonyosnak, hogy a jövő század elejére-közepére várható nanotechnológiai korszak bizonyos értelemben meghosszabbítása lesz a számítógépek és a komputerhálózatok mostani forradalmának: egy olyan éra, amelyben a számítástechnika minden eddiginél nagyobb jelentőségre tesz szert.

Galántai Zoltán

„Minek hirdessek több számítástechnikai lapban is, hiszen ugyanazok olvassák mindegyiket!”

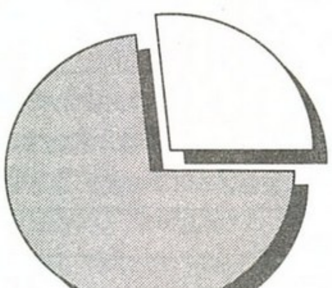
(Ismeretlen klasszikus szerző)

Legfrissebb felmérésünk szerint az Új Alaplap olvasói közül:



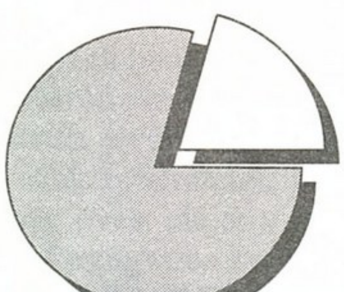
← 27,9% a **Chipet** rendszeresen olvassa

⇐ **...de nem olvassa azt 72,1%**



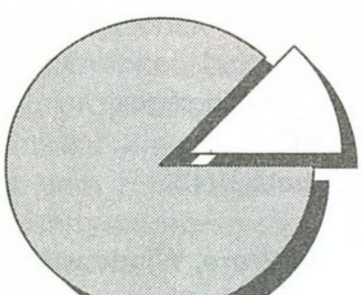
← 26,8% a **PC Worldöt** rendszeresen olvassa

⇐ **...de nem olvassa azt 73,2%**



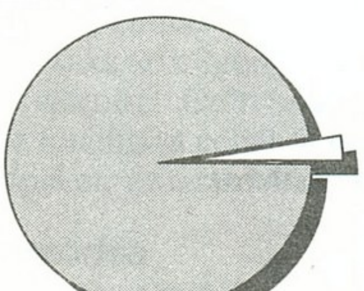
← 20,9% a **Computer Panorámát** rendszeresen olvassa

⇐ **...de nem olvassa azt 79,1%**



← 13,1% a **CW-Számítástechnikát** rendszeresen olvassa

⇐ **...de nem olvassa azt 86,9%**



← 2,8% a **VGA Monitort** rendszeresen olvassa

⇐ **...de nem olvassa azt 97,2%**

Az adatok az Új Alaplap 1997-es olvasói felméréséből származnak.
(Készítette a Mareco közvélemény- és piackutató, a Gallup International magyar tagja.)

Élő-pontszám a számítógépes sakkban is

Célgépek helyett programok

A sakkszámítógépek (mikrogépek) és PC-kre írt programok játékerejének összemérésére a svéd Thoralf Karlsson és néhány társa által alkalmazott mérési rendszer, a „svéd Élő-lista" szakmai körökben általánosan elfogadott mérce, mind a programozók, mind a sakkozók körében. Az egyébként több helyen publikált listát és e cikk egyéb adatait a kéthavonként megjelenő német „Computerschach & Spiele" című német folyóiratból merítettük.

A gépi „sakktudás" összemérésekor alapjában véve azonos számítási módszerről van szó, mint amit a sakkversenyzők erősorrendjének megállapítására alkalmaznak. Ennek megalapozója az USA-ban élt (1903-1992) magyar származású fizikaprofesszor, Élő Árpád.

Ugyanaz — kicsit másként

A FIDE (Nemzetközi Sakkszövetség) által elismert versenyeken elért eredményük alapján mindazok a világranglistára kerülnek, akiknek a pontszáma meghaladja a 2000-et. Az Élő-pontszámokat folyamatosan is vezetik, de hivatalosan félévenként állapítják meg, illetve módosítják azokat, egy-egy versenyző meglévő Élő-pontszámához hozzáadva az előző félévi versenyeken szerzett pontokat, illetve levonva azokat, amelyeket elvesztett. A kiszámításnál a következő szabályszerűség érvényesül:

a) Magasabb a követelmény (az „elvárás") annak teljesítményével szemben, akinek nagyobb a pontszáma.

b) Mindenki aszerint nyer, illetve veszít Élő-pontokat, hogy túlteljesíti az elvárásokat, vagy pedig elmarad azoktól.

Svéd szakembereknek támadt olyan gondolatuk (még a nyolcvanas évek elején), hogy ugyanezt a módszert alkalmazzák a forgalomba került sakkszámítógépek és sakkprogramok játékerejének összehasonlítására.

Eleinte arra törekedtek, hogy a számításhoz minél nagyobb számban gyűjtsenek össze számítógépes játszmákat és versenyeredményeket, beleértve saját játszmáikat is, amelyeket gépekkel, programokkal szemben vívtak.

Csakhamar kiderült azonban, hogy gépek és programok számára nem rendeznek olyan nagy számban versenyeket, hogy versenyeredményeik alapján valós erősorrendet lehessen megállapítani.

Emellett a gépek emberekkel szemben vívott játszmáinak figyelembe vétele talán azért sem nagyon alkalmas arra, hogy reális összhasonlítás készüljön belőle, mert más az, amikor az ember sakkozik (gondolkodik), és más az, amikor a gép játszik (számít). Ezért elhatározták, hogy egy mérvadó ranglista felállítása céljából folyamatosan egymással játszatják a gépeket, programokat.

Átrendeződés

A svéd számítógépes sakkozók egyesületének már 1987-ben 450 tagja volt, és köztük nagy számban olyanok, akik részvételt vállaltak az érdekes tesztelésben.

Az 1987 derekán kiadott lista 29 sakkszámítógép nevét tartalmazza (a PC-kre írt programok ekkor még nem voltak versenyképesek). Ezek sorában több ismerős márkára bukkanhatnak azok az olvasóink, akik hosszabb ideje foglalkoznak sakk-számítógépekkel (Mephisto, Fidelity. Novag, SciSys stb.)

Később az újabb gépek persze alaposan átrendezték a sorrendet, a Budapesten 1983-ban világbajnokságot nyert Fidelity Elite A/S négy év múlva már csak a 22. helyen szerepelt a listán...

Az első táblázat a helyezési sorszám és a gép paraméterei mellett a következő értékeket tartalmazza:

— az aktuális Élő-pontszámot,

— az Élő-pontszám hibahatárát felfelé és lefelé,

— a játszmák számát, amelyeknek alapján az Élő-pontszámot megállapították,

— a játszmákban elért pontok százalékos arányát a játszmák számához képest,

— végül az ellenfelek átlagos Élő-pontszámát.

Az 1. táblázatból kiderül, hogy 1987-ben mintegy ötezer lejátszott parti szolgált a számítás alapjául.

Tíz éve a svédek gyakorlatilag azonos számítási módszerrel, csak éppen egyre több programmal és géppel (a sorrend megfordítása sem véletlen), valamint a mai műszaki fejlettségnek megfelelő eszközökkel folytatják a listák készítését. Ez idő szerint évente hárommal rukkolnak ki. Érdemes megnézni az 1997 júniusában kiadott listát. (2. táblázat.)

Az idei táblázat szerkezete majdnem ugyanolyan, mint a 10 évvel korábbié, mindössze eggyel több rovatot tartalmaz, azt is csak azért, mert a második helyen a lehetséges negatív és pozitív eltéréseket nem szimmetrikusan, hanem mindkét irányban külön jelzik.

A rendszernek talán egyetlen gyengéje, hogy ez az eltérési lehetőség bizony elég nagy, miközben a hasonló teljesítményű programok közötti játékerő különbsége valójában minimális.

Processzorváltás

A rangsor változásait figyelve az ismert programok helycseréinél érdekesebb, hogyan halad előre egy-egy új program, s némelyik hogyan „robban be" az élcsoportba. Igen jelentős fejlődés, hogy a lejátszott partik száma 5000-ről tíz év alatt 55 000-re, a tesztben részt vett programoké és gépeké pedig 29-ről 170-re nőtt, bár a nyilvánosságra kerülő eredménylistán mindig csak az első ötven szerepel.

Az emberek körében a sakkmesterek Élő-pontszámai növekvő tendenciát mutatnak, ami a több versenyből és a sakkozók növekvő számából, illetve jobb játékerejéből következik. A pontszámemelkedés tendenciája ugyanakkor a számítógépes sakkban is megfigyelhető.

A gépi sakkozás — ahogy azt az [Új]]Alaplapban többször is megírtuk — a sakkszámítógépekről (a célgépekről) szinte

teljesen átterelődött a PC-n futtatott programokra. (Egészen más a sakkozó óriásgépek kérdése, de ezzel, a Deep Blue-val és társaival érdemes külön foglalkozni.)

A táblázatokban szereplő programok erőpróbája akkor reális, ha a teszteléshez közel azonos hardvert használnak. A legutóbbi teszteket 90 MHz-es Pentium processzorú gépeken hajtották végre. Egy újabb típus (akár ugyanannak a programnak újabb verziója) csak akkor kerül fel a tesztelési listára, ha már legalább száz játszmát váltottak vele. Most a tesztelésnél folyamatosan áttérnek az Intel MMX 200-as processzor használatára. A különbséget máris jól mutatja, hogy ugyanaz a két program (a Hiarcs és a Rebel) mennyivel jobban „sakkozik" a 200 MHz-es MMX-en.

Felmerülhet a kérdés, hogyan lehet 55 ezer játszmát négy hónap alatt úgy lejátszatni, hogy az eredmény hiteles legyen. A játszmák ugyanis a sakkversenyek szabályzatának megfelelően folynak, vagyis a gépeknek két óra alatt 40, majd óránként 20 lépést kell megtenniük, ami azt jelenti, hogy egy-egy parti akár több órán át is tarthat. Nos, a sakkozógépeket és a sakkprogramokat speciális programok játszatják egymással, csupán indításkor és a játszmák befejezésekor van szükség emberi beavatkozásra. A játszmákat azonban folyamatosan kinyomtatják, és nem publikálják ugyan, de megőrzik azokat, tehát az eljárás szigorúan ellenőrizhető.

Lindner László

1. TÁBLÁZAT

Világranglista 1987

Hely	Sakkszámítógép (processzor)	Pont	+/-	Parti	Nyert	Ellen
1.	Mephisto Dallas 68000 (12 MHz)	2070	36	237	74%	1886
2.	Mephisto Amsterdam 68000 (12 MHz)	2007	26	471	69%	1870
3.	Fidelity Avantgarde (5 MHz)	1921	24	528	57%	1871
4.	Mephisto Rebell (5 MHz)	1899	41	184	49%	1903
5.	Fidelity Par Excellence (5 MHz)	1895	22	608	57%	1848
6.	Novag Constellation Forte (5 MHz)	1874	25	493	54%	1845
7.	Conchess Plymate (5,5 MHz)	1869	21	699	52%	1852
8.	Fidelity Excellence (4 MHz)	1862	23	560	53%	1841
9.	Novag Constellation Expert (4 MHz)	1844	28	400	51%	1833
10.	Conchess Plymate (4 MHz)	1833	30	333	55%	1801
11.	Mephisto Supermondial (4 MHz)	1832	40	193	46%	1860
12.	Mephisto MM II. (3,7 MHz)	1831	40	192	53%	1808
13.	Elegance (3,6 MHz) /Private Line	1829	33	281	58%	1774
14.	SciSys Turbostar 432 (4 MHz)	1825	22	612	51%	1821
15.	Fidelity Excellence (3 MHz)	1824	25	496	52%	1808
16.	Novag Super Constellation (4 MHz)	1782	20	774	45%	1817
17.	Conchess Glasgow (4 MHz)	1771	31	326	50%	1772
18.	Mephisto B+P (3,7 MHz)	1751	69	64	39%	1830
19.	SciSys Superstar 36K (2 MHz)	1734	29	360	44%	1775
20.	Novag Constellation Quattro (4 MHz)	1720	33	278	39%	1802
21.	Fidelity Prestige (4 MHz)	1712	51	117	44%	1754
22.	Fidelity Elite A/S (3,2 MHz)	1705	33	284	40%	1777
23.	Novag Constellation (3,6 MHz)	1696	26	472	41%	1759
24.	Conchess Glasgow (2 MHz)	1694	29	373	40%	1767
25.	Novag Constellation (2 MHz)	1650	30	346	39%	1729
26.	Mephisto Mondial (1,8 MHz)	1632	57	96	45%	1669
27.	Super Enterprise/Advanced Star Chess	1578	42	178	29%	1735
28.	SciSys Turbo 16K (3 MHz)	1514	58	90	29%	1667
29.	SciSys Turbo S-24K (3 MHz)	1512	66	71	23%	1726

2. TÁBLÁZAT

Világranglista 1997

Hely	Program és processzor (vagy gép)	Pont	+	-	Parti	Nyert	Ellen
1.	Hiarcs 6.0 P MMX 200	2587	65	–57	152	73%	2416
2.	Rebel 8.0 P MMX 200	2516	72	–64	118	68%	2382
3.	Hiarcs 6.0 P 90	2470	31	–29	573	65%	2361
4.	Rebel 8.0 P 90	2465	25	–24	877	65%	2358
5.	Hiarcs 5.0 P 90	2443	41	–39	323	63%	2349
6.	MChess Pro 6.0 P 90	2432	24	–24	856	58%	2373
7.	Genius 5.0 P 90	2424	27	–26	690	57%	2376
8.	Rebel 6.0 P 90	2412	31	–30	540	60%	2340
9.	Genius 4.0 P 90	2412	25	–25	809	64%	2313
10.	MChess Pro 5.0 P 90	2411	26	–25	759	62%	2321
11.	Genius 3.0 P 90	2411	27	–26	715	62%	2324
12.	Rebel 7.0 P 90	2406	26	–26	725	60%	2333
13.	Hiarcs 4.0 P 90	2398	26	–25	756	55%	2363
14.	Nimzo 3.5 P 90	2388	30	–29	560	52%	2374
15.	Chessmaster 5000 P 90	2387	49	–45	240	67%	2262
16.	Nimzo 3.0 P 90	2386	26	–25	755	58%	2330
17.	Genius 4.0 486/50–66	2377	26	–26	714	54%	2348
18.	R30 v. 2.5	2374	44	–42	283	66%	2256
19.	Junior 3.3–3.5 P 90	2373	33	–33	436	49%	2377
20.	MChess Pro 4.0 P 90	2360	29	–29	578	52%	2344
21.	Genius 3.0 486/50–66	2359	24	–23	910	62%	2271
22.	Fritz 3.0 P 90	2350	24	–24	855	49%	2354
23.	Fritz 4.0 P90	2337	40	–39	324	60%	2266
24.	MChess Pro 5.0 486/50–66	2336	28	–28	628	52%	2324
25.	Rebel 7.0 486/50–66	2335	26	–26	697	51%	2329
26.	WChess 1.06 P 90	2332	26	–26	738	47%	2354
27.	Kallisto 1.98 P 90	2331	26	–26	697	47%	2353
28.	MChess Pro 4.0 486/50–66	2314	24	–24	846	53%	2290
29.	WChess 1.03 486/50–66	2302	27	–27	677	49%	2308
30.	Rebel 6.0 486/50–66	2300	24	–23	884	55%	2266
31.	Meph Genius 68 030 33 MHz	2299	47	–46	228	57%	2252
32.	Chess Machine 30 MHz King 2.0 aggr/R30 off	2295	21	–20	1190	64%	2192
33.	Chessmaster 4000 486/50–66	2292	34	–33	462	66%	2172
34.	Rebel Decade P 90	2281	29	–30	563	45%	2318
35.	Fritz 3.0 486/50–66	2258	21	–21	1065	50%	2260
36.	Berlin Pro 68 020 24 MHz	2223	25	–24	816	58%	2163
37.	Comet32 P 90	2220	30	–32	558	31%	2357
38.	Kallisto 1.82–1.83 486/50–66	2215	21	–22	1057	42%	2274
39.	Kasparov Sparc 20 MHz	2215	25	–25	769	51%	2207
40.	Saitek RISC 2500 ARM2 14 MHz 128 K	2193	21	–21	1075	56%	2152
41.	Mephisto Montreaux ARM 14 MHz 512 K	2187	32	–30	589	73%	2015
42.	Socrates 3.0 486/33	2141	49	–50	203	47%	2162
43.	Novag Sapphire H8 10 MHz	2086	23	–23	945	49%	2097
44.	Milano Pro SH7000 20 MHz	2077	62	–58	144	62%	1989
45.	Gandalf 2.1 P 90	2041	47	–52	230	27%	2215
46.	Nimzo 2.2.1 486/33	2034	46	–47	229	42%	2088
47.	Zarkov 3.0 486/25–33	2029	46	–48	232	39%	2108
48.	Kasparov Brute Force H8 10 MHz	2018	23	–23	906	44%	2058
49.	Complete Chess System 486/33	1984	47	–47	221	47%	2007
50.	Kasparov President/GK–2100 H8 10 MHz	1976	29	–30	558	47%	1998

MAGYARÁZAT: **Pont** = Élő-pontszám +/– = Az Élő-pontszám lehetséges hibahatára **Parti** = A számítás alapjául szolgáló játszmák száma **Nyert** = A játszmákon szerzett pontok százalékos aránya a játszmaszámhoz viszonyítva **Ellen** = A játszmákban szereplő ellenfelek átlagos Élő-pontszáma

INFORMÁCIÓKÉRÉS: 0932 ▼

QWERTY COMPUTER

Alapítva: 1984-ben

1111 Budapest Bartók Béla út 14.
Tel: 166-9377 (4 vonal) **Fax: 185-2687**
Faxinfo árlistákkal: 166-8292
Internet: http:// www.qwerty.hu
Nyitva: Hétfő - Péntek 10 - 18 óráig

PENTIUM *SZÁMÍTÓGÉPEK TETSZŐLEGES KIÉPÍTÉSBEN, 3 ÉV GARANCIÁVAL*

NOTEBOOK *COMPAQ PORTOCOM* **GSM** *TELEFONOK*

S Z Á M Í T Ó G É P E K   S Z E R V Í Z E

INTERNET *CSATLAKOZZON TELJES SEBESSÉGGEL* QWERTY WEB

WEB lap készítés, üzemeltetés. Előfizetőinknek ingyenes oldal elhelyezés. Internet csatlakozás kiépítése. Nagysebességű Internet kapcsolat vállalkozóknak: **ISDN** bérelt vonal, **routerek** és hálózati konfigurálás.

EPSON, HP, CANON nyomtatók, kiegészítők. **UMAX, EPSON, AGFA** szkennerek.
Multimédia, CD ROM-ok, **CD-írás**, **DTP** rendszerek. **Modemek** viszonteladóknak is, GSM-adatátvitel.
Szoftverek, tartozékok, kiegészítő eszközök, szakkönyvek **SZÉLES VÁLASZTÉKÁVAL VÁRJUK!**

INFORMÁCIÓKÉRÉS: 0920 ▼

http://www.kiskapu.hu

KISKAPU KÖNYVKERESKEDÉS

Angol és magyar nyelvű számítástechnikai szakkönyvek széles választéka!
Szeretettel várjuk önöket nyári akciónkkal, 10 - 20 - 30% árengedmény több angol nyelvű könyvből!

1081 Budapest, Népszínház u. 29.
Tel.: (06-1) 269-9119
fax: (06-1) 303-1619
e-mail: kiskapu@www.kiskapu.hu



VWMacro 1.0

Új taggal bővült a család, és így már a makróvírusok ellen is védelmet nyújt a

VirWare vírusvédelem

Forródrót: (20) 421-174
E-mail: virware@mail.datanet.hu
WEB: www.edu.dote.hu/~virware

...és a biztonság visszatér

INFORMÁCIÓKÉRÉS: 0939 ▼

INFORMÁCIÓKÉRÉS: 0903 ▼

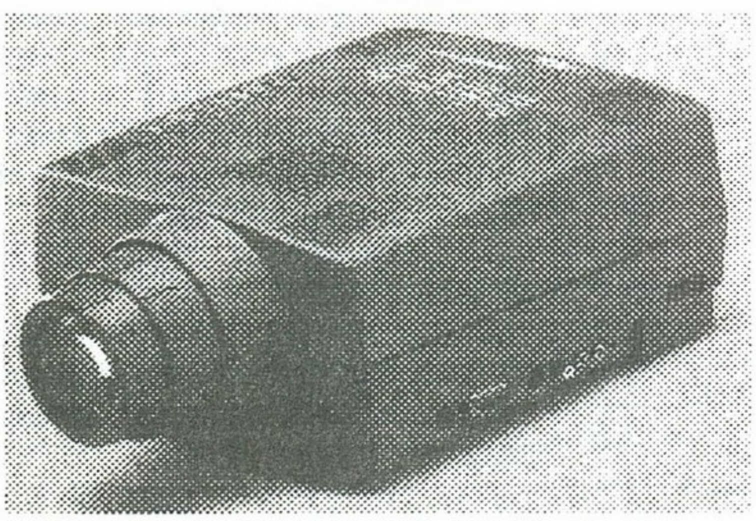
ASK

A világ **legkisebb multimédiás** projektora az **ASK**-tól az

IMPRESSION A4

Súlya **csak 5 kg**, akkora mint egy **A4**-es lap, de **azért tud annyit** mint a nagyok, mert

- felbontása SVGA (800*600)
- fényereje 500 ANSI Lumen
- PC, MAC, video, audio bemenet
- Intelligens távvezérlő



Viszonteladókat keresünk ! Bérlési lehetőség!

array

array Data Hungária Kft.
1094 Bp.,Balázs Béla u. 35.
tel: 455-6892, 455-6893 tel/fax: 455-6894

A Budapesti Oktatásfejlesztési Alapítvány és a Harkály Magániskola egyedülálló lehetőséget biztosít érettségizetteknek államilag elismert

Felsőfokú Számítástechnikai Programozó

végzettség megszerzésére *levelező* szakon. A kb. 2 éves magas színvonalú interaktív oktatási program kényelmes, idő és pénztakarékos megoldást kínál egy keresett szakma megszerzésére. A 11 témakörre bontott képzés költsége 12.000 Ft/témakör. Érdeklődés esetén az alábbi szelvényt kérjük visszaküldeni a ***Harkály Magániskola***
1446 Budapest Pf. 390 címre.

Név: ....

Irsz.: .... **Cím:** ....

....

INFORMÁCIÓKÉRÉS: 0912 ▼

Gráfok és hálózatok kezelése számítógéppel VI.

Hálózatok összefüggősége

Önmagában is érdekes és az alkalmazásokban is gyakran előforduló probléma a gráfok, illetve a hálózatok összefüggőségének vizsgálata. A gráfelméletben többféle „összefüggőség" fogalom is definiálva van. Cikkünkben mi csak egyfélével dolgozunk: összefüggő gráf az olyan gráf, amelyben bármely két pont között van legalább egy út. Ezzel a résszel cikksorozatunk végére értünk. Reméljük, hogy az e témakörben kevésbé jártas érdeklődő diákok, tanárok és gyakorló számítástechnikus kollégák számára sikerült egy kicsit rávilágítanunk erre a fontos szakterületre.

Ha visszagondolunk a sorozat egyik korábbi, a minimális utak keresésével foglalkozó cikkére, azt is mondhatnánk, hogy az összefüggőség nem probléma, hiszen meghatározva a minimális utakat az összes viszonylatra, kiderül, hogy van-e olyan viszonylat, amelynek nincs útja. A dolog természetesen nem ilyen egyszerű; ennek a témakörnek is megvannak a saját speciális feladatai és megoldásai, és az alábbiakban ebből adunk egy kis ízelítőt.

Minimális feszítőfa

Egy összefüggő gráf egy *feszítőfája* a következő értelmezést kapja definíciójaként: a gráfnak egy olyan — fa struktúrájú — részgráfja, amely minden pontot tartalmaz. Ha már hálózatról beszélünk, vagyis az éleknek van hosszuk, akkor egy ilyen fát is minősíthetünk, mérhetünk az élei hosszának összegével. Így már beszélhetünk két ilyen fa hossz szerinti összehasonlításáról, és kereshetjük a legrövidebb, a minimális hosszúságút. (Megjegyezzük, hogy ilyenből egy hálózaton belül több is lehet.) A *minimális hosszúságú feszítőfát* a továbbiakban röviden *minimális feszítőfának* nevezzük.

A keresés bonyolultsága lényegesen megnő akkor, ha a gráf irányított. Ezért itt és most csak az egyszerűbb esettel foglalkozunk: a hálózat irányítatlan, más szóval minden él kétirányú, és a két irányban azonos hosszú. Az 1. ábra hálózata ilyen, a vastagított élek egy minimális feszítőfát adnak.

Szemléltessük egy példával a feladatot. Egy település utcahálózatának kereszteződései legyenek a pontok, az összekötő szakaszok az élek. Az önkormányzat a jelenlegi utcahálózat bizonyos szakaszain (mindkét irányban használható) kerékpárutat akar kiépíteni. A kiépítés költsége a hálózat minden szakaszára ismert. Mely szakaszokon építsenek kerékpárutat, ha be kell tartani az alábbi követelményeket:

— Minden pontból minden pontba el lehessen jutni csak a kerékpárutakat használva.

— A kiépítés költsége a lehető legkisebb legyen.

A megoldás: a hálózatban az élhossz legyen minden élen a kerékpárútnak a megfelelő útszakaszon való kiépítési költsége, és határozzunk meg egy minimális feszítőfát. Ennek élei mentén kell kiépíteni a kerékpárutakat. (Bizonyára észrevette mindenki, hogy itt hallgatólagosan elfogadtuk azt a tényt — ami egyébként egzaktan bizonyíthatóan igaz —, hogy a minimális összélhosszúságú, és minden pontot tartalmazó részgráf szükségszerűen egy fa.)

Az ismertetendő (Kruskal-féle) algoritmus a hálózat pontjaiból képez először egyelemű, majd minden lépésben bővülő halmazokat. Ezek a halmazok önmagukban összefüggő és minimális összhosszú részhálózatoknak felelnek meg.

Két ilyen halmazt akkor tudunk egyesíteni, ha van a két részhálózatot összekötő él. Az élek vizsgálatát hossz szerinti növekvő sorrendben végezzük. Ha sikerül az összes pontot egy halmazba összeszedni, akkor a gráf összefüggő, és az egyesítéseknél „összekötő" élek egy minimális feszítőfát alkotnak.

Az algoritmus pontosabb leírásához vezessünk be néhány jelölést:

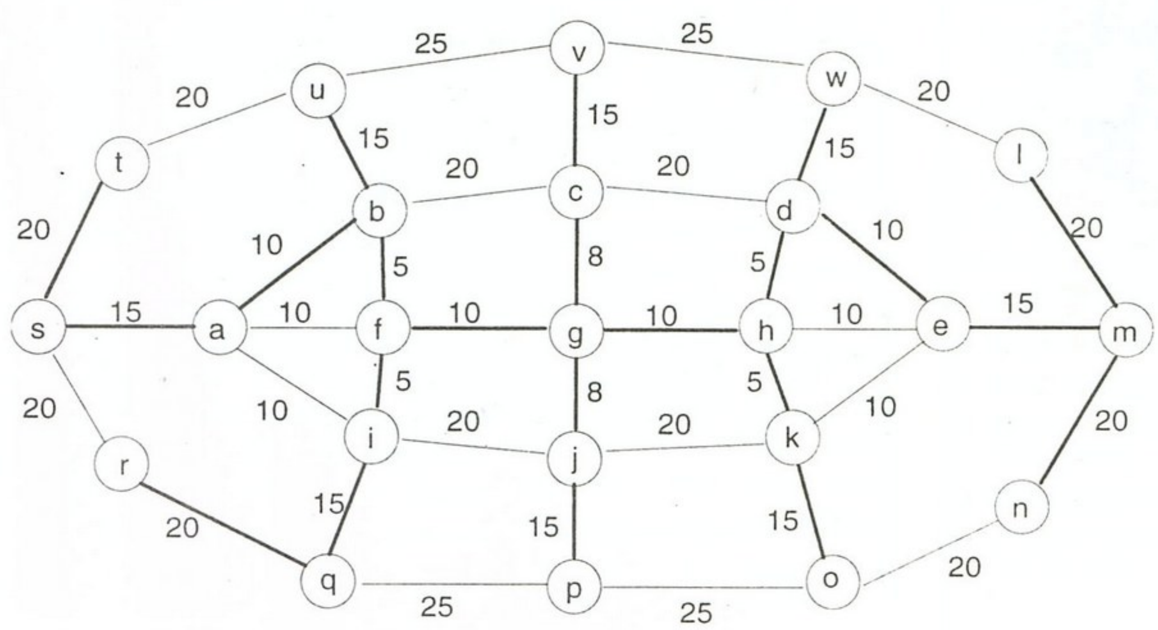
— xy jelölje az x és y pontokat összekötő élet.

— E jelölje az élek egy részhalmazát. Ez kezdetben az összes élt tartalmazza. Ebből választjuk ki sorra a vizsgálandó éleket.

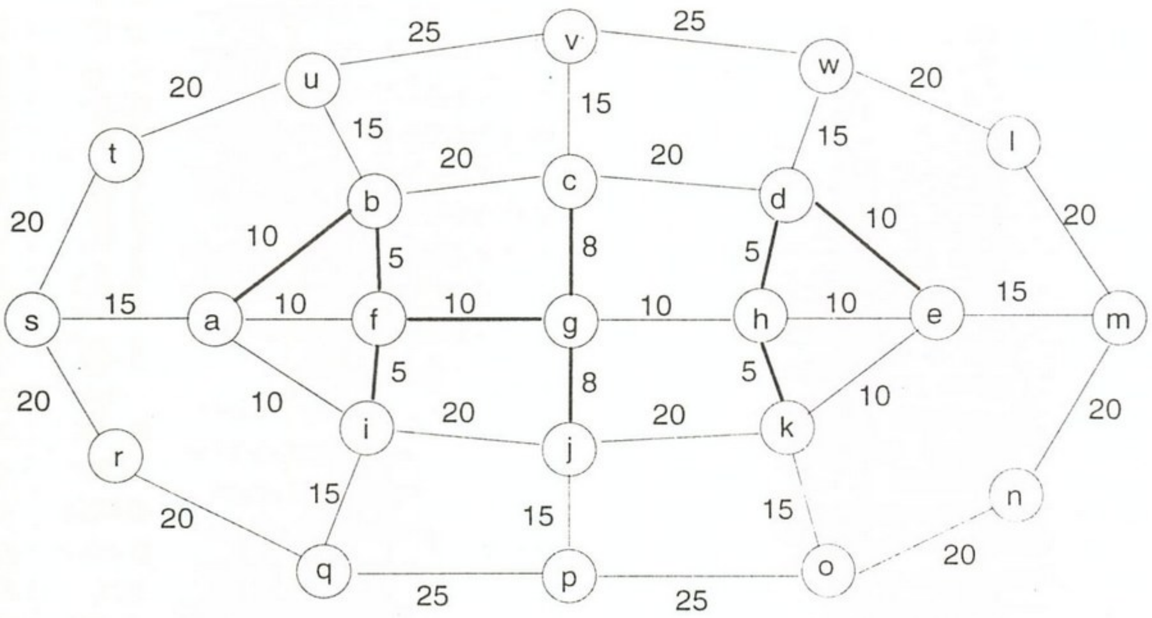
— F jelölje az élek egy részhalmazát. Ez kezdetben üres. Ebbe gyűjtjük a fát alkotó éleket.

— H jelöljön egy olyan halmazt, amelynek elemei is halmazok, mégpedig ponthalmazok. A H elemei önmagukban összefüggő és minimális összhosszú részhálózatoknak felelnek meg.

Ezek után az algoritmust a következőképpen definiálhatjuk:



1. ábra



2. ábra

Kezdőállapot

E: minden élt tartalmaz. *F:* üres. *H:* minden egyelemű ponthalmazt tartalmaz.

Javító lépések

Mindaddig, amíg a *H*-ban legalább két halmaz van, és emellett az *E*-ben is van legalább egy él, ismételjük az alábbi műveletsort:

Válasszunk ki és töröljünk is az *E*-ből egy minimális hosszú élet, jelölje ezt *xy*. Ha az *x* és az *y* a *H*-ban lévő két különböző halmazban van, akkor egyrészt az *xy* élet vegyük fel az *F*-be, másrészt a két halmazt egyesítsük, az egyesített halmazt vegyük fel a *H*-ba, az eredetieket pedig töröljük a *H*-ból.

Végállapot

Ha a javító lépéseket azért nem tudjuk folytatni, mert a *H*-ban csak egy halmaz van, akkor készen vagyunk, az *F* adja a megoldást. Ha viszont a befejezés oka az, hogy még több halmaz van ugyan, de nincs több él az *E*-ben, akkor a hálózat nem összefüggő, nincs feszítőfa.

Az algoritmust a 3. ábrán követhetjük a példahálózaton. Megadjuk a kezdőállapotot, az első három javító lépést, egy közbeeső javító lépést, majd a végállapotot. A közbeeső lépés utáni fát a 2. ábra, a végeredményt az 1. ábra is mutatja. (A példában az *E* halmaz nem szerepel.) Meg kell jegyeznünk még, hogy a minimális hosszú él kiválasztásánál — ha erre több lehetőség is volt — mindig az ábécésorrend szerinti elsőt választottuk ki. (Ez természetesen nem előírás, csak egy lehetőség. Máshogy választva esetleg egy másik minimális feszítőfát kaphatunk.)

Összefüggő komponensek

A témakör egy másik alapfeladata a következőképpen fogalmazható meg: „Bontsuk fel a hálózatot önmagukban összefüggő részhálózatokra, más néven komponensekre."

Bár a feladat látszólag más, mint az előbb, de rögtön látni fogjuk, hogy tulajdonképpen a fenti feladat egy egyszerűbb speciális esetéről van szó, amelyet az előbbi algoritmus egy szintén egyszerűsített változatával oldhatunk meg.

Egyszerűsítsük az algoritmust! Mivel itt nincs szó hosszakról (tehát a hálózat csak mint gráf játszik szerepet), az élek vizsgálati sorrendje közömbös. Fára mint eredményre nincs szükségünk, tehát az éleket nem kell gyűjteni. Az algoritmus ezek után:

Kezdőállapot

E: minden élt tartalmaz. *H:* minden egyelemű ponthalmazt tartalmaz.

Javító lépések

Mindaddig, amíg a *H*-ban legalább két halmaz van, és emellett az E -ben is van legalább egy él, ismételjük az alábbi műveletsort:

Válasszunk ki és töröljünk is az *E*-ből egy élet, jelölje ezt *xy*. Ha az *x* és az *y* a *H*-ban lévő két különböző halmazban van, akkor a két halmazt egyesítsük, az egyesített halmazt vegyük fel a *H*-ba, az eredetieket pedig töröljük a *H*-ból.

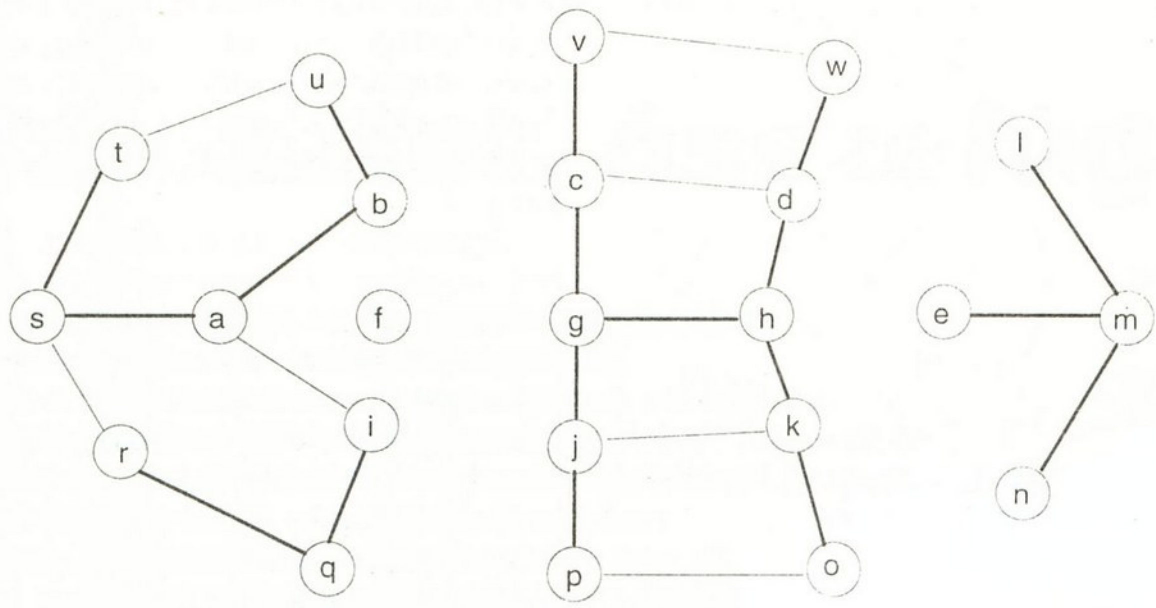
Végállapot

A *H*-ban lévő halmazok adják az összefüggő komponenseket. (Ha csak egy ilyen van, akkor a hálózat összefüggő.)

Az algoritmus szemléltetésére „szétszedtük" a példahálózatot, töröltük az élhosszakat, így kaptuk a 4. ábra gráfját. Feltételezve, hogy a gráfot az éltárolásos rendszerben tároljuk, a legegyszerűbb az élek vizsgálatát is ebben a sorrendben végezni. A *H* változását az 5. ábrán követhetjük. Az ábrán soronként az egy-egy kezdőpontból kiinduló összes él vizsgálata utáni állapotot adjuk meg.

F																							
H	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
F	bf																						
H	a	b,f		c	d	e	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
F	bf	dh																					
H	a	b,f		c	d,h		e	g	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
F	bf	dh	fi																				
H	a	b,f,i			d,h		e	g	c	j	k	l	m	n	o	p	q	r	s	t	u	v	w
	. . .																						
F	bf	dh	fi	hk	cg	gj	ab	de															
H	a,b,f,i				d,h,k,e			g,c,j				l	m	n	o	p	q	r	s	t	u	v	w
F	bf	dh	fi	hk	cg	gj	ab	de	fg														
H	a,b,f,i,g,c,j							d,h,k,e				l	m	n	o	p	q	r	s	t	u	v	w
	. . .																						
F	bf	dh	fi	hk	cg	gj	ab	de	fg	gh	as	bu	cv	dw	em	ko	jp	iq	lm	mn	qr	st	
H	a,b,f,i,g,c,j,d,h,k,e,s,u,v,w,m,o,p,q,l,n,r,t																						

3. ábra



4. ábra

Programtechnika

A fenti algoritmusok beprogramozásánál több olyan nehézségre is bukkanunk, amelynek megoldása, áthidalása nem nyilvánvaló, sőt komoly programtervezési meggondolásokat is igényel. Ilyen lehet például a halmazok (*E, F, H*) konkrét megvalósítása, ábrázolása — különösen, ha nagyobb hálózatokra is gondolunk. Ezért a programszövegek megértését itteni magyarázattal is segítjük.

Kezdjük talán a *H* halmaz tárolásával/kezelésével. Ez a halmaz olyan, egymással diszjunkt (közös elem nélküli) ponthalmazokból áll, amelyek uniója (egyesítése) lefedi/magába foglalja a gráf összes pontját. Másképpen fogalmazva: a gráf valamennyi pontjára igaz az, hogy benne van *H*-nak valamely ponthalmazában (hogy egy adott pont éppen melyik halmazban van, azt az algoritmus lépései definiálják), így minden pontra csak ezt az információt kell feljegyeznünk (azaz, hogy éppen melyik halmazban található).

Mivel a *H*-beli ponthalmazok száma (jelölje ezt *HDB*)

— kezdetben megegyezik a pontok számával,

— az algoritmusok során mindig csökken (hiszen halmazokat egyesítünk),

— egynél nem lehet kevesebb,

ezért, ha megsorszámozzuk a *H*-beli halmazokat, és ezt a halmazsorszámot rendeljük hozzá a pontokhoz, akkor egy pontszám elemű tömbben tárolható a *H* halmaz.

A kérdés ezek után már csak az, hogyan adminisztráljuk a *H* halmaz megváltozását, azaz két *H*-beli ponthalmaz egyesítését/összevonását. Ahhoz, hogy a *H*-beli halmazok megtartsák 1-től induló folyamatos sorszámozásukat, a nagyobb sorszámú összevonandó halmaz elemeit áttesszük a kisebb sorszámú halmazba, míg ezen „kiürülő" (nagyobb sorszámú) halmazt úgy töröljük, hogy a legutolsó (*HDB* sorszámú) halmaz elemeit átrakjuk ebbe a halmazba. Ezek a „pontmozgatások" egyetlen ciklussal megtehetők, hiszen csak a „mozgatandó" pontokhoz rendelt halmazsorszámokat kell megváltoztatnunk, és ezután már csak a *H*-t alkotó ponthalmazok számát (*HDB*) kell eggyel csökkentenünk.

Az *E* és *F* élhalmazok kezelése ennél egyszerűbben megoldható, hiszen egy élről csak azt kell nyilvántartanunk, hogy az algoritmus során éppen melyik halmazban található (*E*-ben, *F*-ben, vagy egyikben sem). Ezt az információt pedig a gráf tárolási módjától függően vagy egy egydimenziós tömbben (éltárolás), vagy egy kétdimenziós tömbben/mátrixban (mátrixos tárolás) tárolhatjuk/kezelhetjük.

Témazáró

Továbbolvasásra, a témakörben való mélyebb tájékozódáshoz ajánlunk néhány magyar nyelvű és bevezető, tankönyv jellegű szakkönyvet:

Wirth: Algoritmusok + Adatstruktúrák = Programok

Aho — Hopcroft — Ullman : Számítógép — algoritmusok tervezése és analízise

Lovász — Gács: Algoritmusok

Mindhárom könyv a Műszaki Könyvkiadó (Budapest) többszöri kiadásában is megjelent.

Marton László — Pusztai Pál

A sorozatnak ehhez a részéhez is készült a lemezmellékletre illusztráció, de anyagtorlódás miatt oda most nem fért fel, ezért csak következő számunk lemezmellékletén találhatják majd meg. (A szerk.)

<table>
<tr><td>H</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td><td>g</td><td>h</td><td>i</td><td>j</td><td>k</td><td>l</td><td>m</td><td>n</td><td>o</td><td>p</td><td>q</td><td>r</td><td>s</td><td>t</td><td>u</td><td>v</td><td>w</td></tr>
<tr><td>H</td><td colspan="4">a,b,i,s</td><td>e</td><td>f</td><td>g</td><td>h</td><td>c</td><td>j</td><td>k</td><td>l</td><td>m</td><td>n</td><td>o</td><td>p</td><td>q</td><td>r</td><td>d</td><td>t</td><td>u</td><td>v</td><td>w</td></tr>
<tr><td>H</td><td colspan="5">a,b,i,s,u</td><td>f</td><td>g</td><td>h</td><td>c</td><td>j</td><td>k</td><td>l</td><td>m</td><td>n</td><td>o</td><td>p</td><td>q</td><td>r</td><td>d</td><td>t</td><td>e</td><td>v</td><td>w</td></tr>
<tr><td>H</td><td colspan="5">a,b,i,s,u</td><td>f</td><td>g</td><td>h</td><td>c</td><td>j</td><td>k</td><td>l</td><td>m</td><td>n</td><td>o</td><td>p</td><td>q</td><td>r</td><td>d</td><td>t</td><td>e</td><td>v</td><td>w</td></tr>
<tr><td>H</td><td colspan="5">a,b,i,s,u</td><td>f</td><td>h</td><td colspan="4">c,d,g,v</td><td>l</td><td>m</td><td>n</td><td>o</td><td>p</td><td>q</td><td>r</td><td>j</td><td>t</td><td>e</td><td>k</td><td>w</td></tr>
<tr><td>H</td><td colspan="5">a,b,i,s,u</td><td>f</td><td colspan="6">c,d,g,v,h,w</td><td>m</td><td>n</td><td>o</td><td>p</td><td>q</td><td>r</td><td>j</td><td>t</td><td>e</td><td>k</td><td>l</td></tr>
<tr><td>H</td><td colspan="5">a,b,i,s,u</td><td>f</td><td colspan="6">c,d,g,v,h,w</td><td>k</td><td>n</td><td>o</td><td>p</td><td>q</td><td>r</td><td>j</td><td>t</td><td colspan="2">e,m</td><td>l</td></tr>
<tr><td>H</td><td colspan="5">a,b,i,s,u</td><td>f</td><td colspan="7">c,d,g,v,h,w,j</td><td>n</td><td>o</td><td>p</td><td>q</td><td>r</td><td>k</td><td>t</td><td colspan="2">e,m</td><td>l</td></tr>
<tr><td>H</td><td colspan="5">a,b,i,s,u</td><td>f</td><td colspan="8">c,d,g,v,h,w,j,k</td><td>o</td><td>p</td><td>q</td><td>r</td><td>n</td><td>t</td><td colspan="2">e,m</td><td>l</td></tr>
<tr><td>H</td><td colspan="6">a,b,i,s,u,q</td><td>f</td><td colspan="8">c,d,g,v,h,w,j,k</td><td>o</td><td>p</td><td>r</td><td>n</td><td>t</td><td colspan="2">e,m</td><td>l</td></tr>
<tr><td>H</td><td colspan="6">a,b,i,s,u,q</td><td>f</td><td colspan="9">c,d,g,v,h,w,j,k,p</td><td>o</td><td>r</td><td>n</td><td>t</td><td colspan="2">e,m</td><td>l</td></tr>
<tr><td>H</td><td colspan="6">a,b,i,s,u,q</td><td>f</td><td colspan="10">c,d,g,v,h,w,j,k,p,o</td><td>r</td><td>n</td><td>t</td><td colspan="2">e,m</td><td>l</td></tr>
<tr><td>H</td><td colspan="6">a,b,i,s,u,q</td><td>f</td><td colspan="10">c,d,g,v,h,w,j,k,p,o</td><td>r</td><td>n</td><td>t</td><td colspan="3">e,m,l</td></tr>
<tr><td>H</td><td colspan="6">a,b,i,s,u,q</td><td>f</td><td colspan="10">c,d,g,v,h,w,j,k,p,o</td><td>r</td><td>t</td><td colspan="4">n,e,m,l</td></tr>
<tr><td>H</td><td colspan="7">a,b,i,s,u,q,r</td><td>f</td><td colspan="10">c,d,g,v,h,w,j,k,p,o</td><td>t</td><td colspan="4">n,e,m,l</td></tr>
<tr><td>H</td><td colspan="8">a,b,i,s,u,q,r,t</td><td>f</td><td colspan="10">c,d,g,v,h,w,j,k,p,o</td><td colspan="4">n,e,m,l</td></tr>
</table>

5. ábra

E SZÁMUNK HIRDETŐI

Cég	Info#	Old.
Allegro	0940	24.
Areco	0902	62.
Array Data	0903	53.
Axico	0927	61.
Cartographia	0904	41.
ComputerBooks	0905	41.
Crown-Tech	0906	42.
Delphi-Szoft	0907	04.
Digital	0901	B4.
DIT Digitáltechnika	0908	34.
EuroWeb	0909	22.
FAN	0910	44.
G70	0911	04.
Harkály Magániskola	0912	53.
Hunix	0913	K4.
IBM	0914	B3.
Infopen	0915	50.
Intergraph	0916	42.
InterPC Networking	0917	62.
Keszo	0918	K4.
Kim-Soft	0919	50.
Kiskapu	0920	53.
Kvint-R	0921	22.
Lias-Networx	0922	43.
MP Computer	0923	K4.
Next	0924	22.
Onyx	0925	B2.
Open Gates	0926	21.
PC Szoftver	0928	61.
Profi Plusz 2000	0929	41.
Profon	0930	61.
Qwerty	0931	34.
Qwerty	0932	53.
Reflex	0933	02.
Rich Selling	0934	02.
Samsung	0935	02.
SCI-Modem	0936	50.
Server	0937	42.
Shartech	0938	61.
Shift Informatika	0939	53.
Spieler	0941	34.
Telelogic	0942	61.
Telnet	0943	62.
Teta	0944	34.
Voxer	0945	04.

A Mikrobazár rovatban a nem kereskedelmi célú egyéni hirdetések közlése ingyenes.

A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint.

A terjedelem alapján így kiszámított összeget kérjük átutalni az Új Alaplap Kiadói Kft számlájára (OTP, 11706016–20788599), vagy feladni postai utalványon a kiadó címére (1539 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés". A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

Szerzői jogokat sértő szoftverhirdetéseket nem közlünk le.

Bármilyen típusú szöveg fordítását vállalom angolról magyarra, magyarról angol nyelvre, illetve vállalom kiadványok látványtervezését, szerkesztését is. Cím: Lachner Zoltán, 1195 Budapest XIX., Jáhn Ferenc u. 14/a. Telefon: 157-0308.

OBJECTS 2.0 — objektumorientált programozás CLIPPER-ben. Tájékoztató kérhető az alábbi címen: Szűcs János, 4400 Nyíregyháza, Vasvári Pál u. 37. Tel.: (42) 437-331 vagy 465-666/1382-es m.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is vállalok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV/33.

Alaplapcsere, memória-, winchester- és floppybővítés a helyszínen. MegaSoft. Telefon: 295-5085.

Stúdiómban megbízhatóan, ellenőrzötten lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hard- ver- és szoftverleírásait. Áfás számlát állítok ki. Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 168-4874.

QuarkXpress, Freehand és Photoshop ismeretekkel, kolormontírozói gyakorlattal, jogosítvánnyal rendelkező 38 éves **nyomdász** bejárós munkát vállal. Telefon: 403-4304.

Német-magyar nyelvről-nyelvre bármilyen szöveg fordítását, szerkesztését vállalom. Telefon: Vati Eszter 140-0046.

Akarod, hogy tiéd legyen ingyen az évszázad viccgyűjteménye? Nos ha igen, akkor írj az alábbi e-mail címre: qbolusyj @gold.uni-miskolc.hu, vagy küldj egy üres kislemezt és egy felbélyegzett válaszborítékot a postacímemre: Kovács Gábor, 3502 Miskolc II., Pf. 83.

Nincsen számítógépe? (Vagy nem tudja használni?) Szívesen szövegszerkesztek bármilyen szöveget bárkinek. Érdeklődni: Votisky András, 121-5887 (egész nap).

Aki tudja, hogyan használható az MS-DOS 25h megszakítása merevlemezre is (vagy hogyan helyettesíthető), azt kérem, árulja el nekem is. Telefon: Hódi Gyula, 221-7289.

Eladó egy Amiga 500-as számítógép tartozékokkal (színes monitor, egér, joystick). Érdeklődni: (29)372-183.

Szeretném felvenni a kapcsolatot minél több programcserélő csoporttal, programozási körrel. Kérem, aki tagja vagy szervezője ilyennek, küldjön információt a csapatról. Címem: Tausz Krisztián, 7988 Darány, Rákóczi F. u. 17/1.

Tehetséges, építőmérnöknek készülő, közlekedési építő-számítástechnika-informatika szakos, végzős szakközépiskolás fiamnak **ösztöndíjat keresek**. Telefon: (76)352-066.

Emlékeztető a feledékenyeknek!

1997. július 1-jével az Új Alaplap szerkesztősége és kiadója elköltözött. Kérjük azokat, akiknek az adatbázisában, címlistájában, telefonjegyzékében szerepeltünk, hogy adatainkat az alábbiak szerint javítsák át:

	Eddig:	**Ezentúl:**
Címünk helyileg:	Bp. I., Márvány u. 17.	**Bp. VI., Dózsa György út 84/b, I. em. 29.**
Közvetlen telefon:	(Nem volt)	**322-4417, 322-5238**
Telefonközpont:	156-3211/ 200, 201, 214	**351-0690/ 242, 243, 244**
Webcím:	(Nem volt)	**http://www.alaplap.hu**

Változatlanok alábbi címadataink:

Postacím:	**1539 Budapest, Pf. 571** (Egy fix pont a sok változás közepette.)
Közvetlen fax:	**214-9492** (Vagy marad, vagy változik, ha egyszer lezajlik a vonaláthelyezés.)
E-mail:	**alaplap@mail.datanet.hu** és **alaplap@telnet.hu** (Mindkettőt megtartjuk.)

Szoftverfejlesztés

Aki profi akar lenni...

A most bemutatott két új könyv a maga módján jó segítséget jelenthet mindazoknak, akik profi szintű programokat szeretnének készíteni, véres verejtékezés, vagy mázsányi szakirodalom állandó lapozgatása nélkül is.

Gary Cornell:

Delphi

Tippek és trükkök
az 1. és 2. verzióhoz,
16, ill. 32 bites fejlesztésekhez
Panem — McGraw-Hill, 1997
436 oldal, 1990 Ft, kemény kötésben.
(A felsőoktatási tankönyvpályázat keretében támogatott mű.)

Ez a könyv a hagyományos Turbo Pascal programozóknak mutatja be, hogy ugyanolyan (vagy akár kevesebb) fáradsággal tökéletes Windows programokat is gyárthatnak, ha átlépnek a Delphi világába.

Annak idején a Borland cég forradalmi változást hozott a korszerű fordítóprogramok elterjesztésében: a Turbo Pascallal a „laikusok" kezébe adta a programfejlesztés összkomfortos, máig páratlanul kényelmesnek és hatékonynak számító keretrendszerét. Az újdonsült programozók jó része azóta elérte vagy túl is haladta a profi programozók szintjét. A újabb változatokkal már objektumorientált programokat is készíthettek, sőt a Turbo Pascal for Windows megjelenésével a grafikus képernyős fejlesztések előtt is megnyílt az út. Hagyományos módszerekkel azonban „igazi" Windows-programok készítésébe belevágni istenkísértés, még a Windows API függvények birtokában és ismeretében is. (Egyedül az API függvények hívásának standard dokumentációja kb. 2000 oldalt tesz ki!)

Könnyen, gyorsan — windowsul is

A Delphi, a Turbo Pascal egyenes ági leszármazottja nem negligálja azt a hatalmas tudást, amit az API függvények megtestesítenek. Használni tudja őket, nem mindig van rájuk szüksége: alig van olyan programozási feladat, amelyet beépített függvényeivel, eljárásaival a Delphi ne tudna ezek nélkül is sokkal egyszerűbben és kényelmesebben megoldani.

A Delphi kevésbé tapasztalt programozóknak is jóformán ugyanolyan hatékonyságot biztosít, amilyet eddig csak a C/C++ használatával lehetett elérni, de ugyanolyan könnyedséget és kényelmet is, amilyet eleinte csak a Visual Basic tudott nyújtani, ráadásul mindezt a Turbo Pascalban megszokott rugalmasággal. Végül — hogy a grafikus fejlesztések legfontosabb gyakorlati szempontjáról se feledkezzünk meg — ugyanazt a biztonságot is megadja az ezerszer bonyolultabb Windows programok fejlesztésében, amit a „sima" DOS programoknál viszonylag könnyű volt elérni. A Delphi ugyanis a C++-ban bevezetett kivételkezelés módszerének frappáns alkalmazásával „előre látja", mikor fenyeget a rendszerhalál réme, és figyelmeztet rá, mielőtt az bekövetkezne.

Az új verzió

Különös módon a Delphinek már az 1.0 változata is kiérlelt, gyakorlatilag hibamentes termékké sikeredett. Nem a hibák kijavítása indokolta tehát a 2.0 verzió elkészítését (mint sok cég szoftvertermékénél megszokhattunk), hanem főleg az, hogy jobban ki lehessen aknázni az új processzorok technikai lehetőségeit.

Az új változatnak természetesen számos előnye van. Nagy előrelépést hozott a Delphi alatt készített alkalmazások sebességének jelentős növelésével, az újrafelhasználható objektumok és komplett DLL könyvtárak létrehozásának fokozott támogatásával, valamint a memóriafoglalás és memóriafelszabadítás további automatizálásával. A Delphi 2.0 lehetővé teszi OLE támogató alkalmazások fejlesztését, majdhogynem korlátlan hosszúságú sztringek könnyű kezelését, sőt a beépített TThread komponens segítségével párhuzamosan futtatható programrészek készítését is. Már előzőleg is hihetetlen sebességgel dolgozott a Delphi fordítója — most az új változat számára mégis átírták, egyrészt optimalizálási megfontolásokból, másrészt hogy informatívabb üzeneteivel jobban segítse a programozókat.

Írni vagy fordítani?

A fentiek tükrében nagyon hasznos tehát, hogy magyarul is elérhetővé vált egy friss információkat nyújtó könyv az új változatról. A könyv szerzője, az amerikai Gary Cornell azokra az olvasókra is tekintettel van, akik egyelőre nem tudnak belépni a 32 bites processzorok paradicsomába. Párhuzamosan tárgyalja mindkét verzió lehetőségeit, sőt arra is figyelmeztet, hogy a Delphivel készülő alkalmazások felhasználói között is lehetnek olyanok, akik csak 16 bites hardverrel rendelkeznek. (Érdemes hát meggondolnunk, hogy az ő kedvükért ne szorítkozzunk-e a két verzió azonos módon kezelhető komponenseire.)

Azokra a magyar olvasókra természetesen nem gondolhatott, akik nem tudják teljesíteni gyakran felhangzó tanácsát, hogy ezt vagy azt el kell olvasni az online dokumentációból — például azért, mert éppen ebből a könyvből szeretnék megismerni a Delphit, mielőtt a szoftvert megvennék. A szerző alapállása érthető, de a magyar kiadásban talán jobb lett volna pótolni legalább az alapvető dolgokat a Borland dokumentációja alapján. (Különös tekintettel arra, hogy tankönyvnek is szánták.)

Csábító a könyv sorozatjelző alcíme („tippek és trükkök"), de nem igazán illik a mi fogalmaink szerinti tippekre és trükkökre. Az eredeti angol kifejezés (Nuts & Bolts = „csavarok" és „anyák") a nyomdatechnikailag is kiugrasztott részekre utal, tartalmukat illetően azonban csak ritka kivételképpen találó szó a „tipp". Ezek egyszerű megjegyzések a Visual Basicet vagy a Pascalt ismerő olvasók számára.

Szénakazal effektus

Nagyobb baj az, hogy a könyv szövegéből nem kerekednek ki eléggé a Delphi előnyei. A könnyed esszéstílus

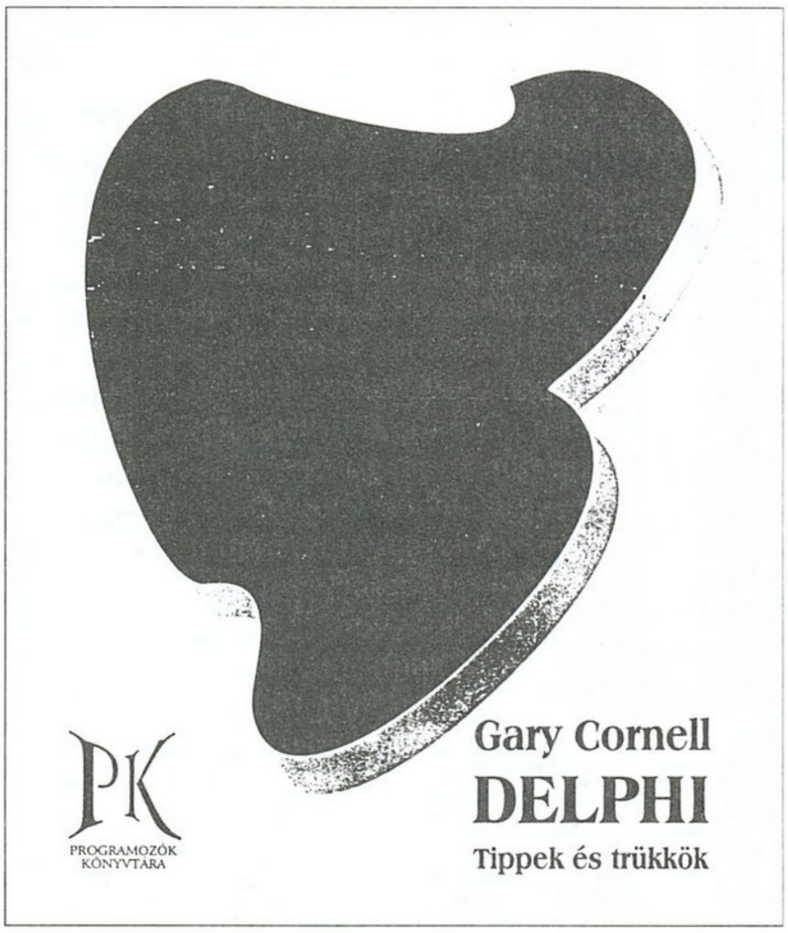
tetszetős ugyan, de nem elég alapos, a legprecízebb hivatkozás sem megy mélyebbre a fejezet megjelölésénél, és ami különösen zavaró, beváltatlan ígéretek is előfordulnak a szövegben. (A megadott fejezetben például hiába keressük, hogyan valósítható meg a TMemo komponens segítségével fájlban tárolt adatok gyors beolvasása, vagy hol ír részletesebben az elrendezéskezelő palettáról, pedig több helyen is megígéri.) Mindezt ellensúlyozhatná egy jobban kidolgozott és teljesebb tárgymutató, így azonban sok témára hiába keresünk. A megadott lapszámok gyakran nem a lényeges dolgokra utalnak (például a Dialógusablak vagy az Objektumkonverzió esetében), sok fontos téma pedig teljesen hiányzik a tárgymutatóból (így a Felbontás beállítása, a Beépített konstansok vagy a Virtuális metódusok).

A könyv előnyére válna, ha sokkal markánsabban kiemelné a szerző a Delphi előnyös tulajdonságait. Ezek közé tartozik például a grafikus felbontás menet közben való állíthatósága, amely most a Screen objektum egyik tulajdonságaként mint „Scaled" eléggé elsikkad a sok egyéb részlet között. Nem segíti az áttekinthetőséget a fejezeteken belüli hierarchia sem, mert a nyomdatechnika ezt nem szolgálja kellőképpen. Szinte érthetetlen, hogy miért kellett három hierarchiaszinten azonos méretű betűkkel jelölni az alcímeket. Számozással is kiemelve segíteni lehetett volna a hivatkozások lazaságán is, függetlenül attól, hogy az angol eredetiben hogy van!

Tisztába tenni a gyereket

Az általános értelemben vett objektumok, a komponensek és a vezérlőelemek a könyvben pontosabb megkülönböztetést érdemelnének. Nem eléggé világos például az alábbi megjegyzés: „A dokumentáció megkülönbözteti egymástól a látható komponenseket és a vezérlőelemeket (controls). A vezérlőelemek azok a komponensek, amelyeket a felhasználó lát." Ebből nem derül ki egyértelműen, hogy a felhasználó itt nem az alkalmazói program fejlesztője, hanem a kész program felhasználója, és hogy mikor mi az, ami látható. A Delphi legfontosabb újításai közé tartozik a komponensek fogalmának bevezetése, és az új koncepció számítástechnikai realizálása, ezeken a kérdéseken tehát nem lenne szabad ilyen könnyedén átsiklani.

A Delphi az Objektum elnevezést használja azokra a TObject osztályból származtatott általános objektumokra, amelyek nem komponensek (TBitmap, TCanvas, TClipboard, TIcon stb.). Ezek csak akkor érhetők el, ha a Create metódussal már létrehoztuk őket, ebben különböznek a komponensektől. A Komponensek olyan előre elkészített, speciális objektumok, amelyek fejlesztési vagy futási időben elérhetők, és absztrakt építőkövekként (de más alakban!) felhasználhatók a programfejlesztésben. A „sima" komponensek legtöbbje közvetlenül levehető valamelyik palettáról, és ikonján keresztül beleépíthető egy „műszerfalba" (például a különböző dialógusablakokba, a főmenübe, a felbukkanó menübe stb.). Ami fontos: a komponensek tulajdonságai fejlesztési időben is, futtatáskor is szabadon gyúrhatók, változtathatók, és a változtatást a Delphi azonnal végrehajtja.



Más komponensek is vannak, amelyek nem szerepelnek ugyan a palettákon, mégis elérhetők, mert mindig van egy hívószámuk (Tag tulajdonság), és az adott alkalmazáson belül egy sorszámuk (ComponentIndex tulajdonság), ez utóbbi az alkalmazás képernyőjén keresztül történő megjelenítéshez. Ilyen komponens reprezentálja például a menüelemeket, rekordmezőket és a képernyőállapotot.

Kissé elkülönülnek a „közönséges" komponensektől a Vezérlő Elemek. Ezek olyan speciális komponensek, amelyek fejlesztési időben és futtatási időben azonos módon jelennek meg (nyomógombok, listák és egyéb ablakelemek, címkék, különböző alakzatokat megjelenítő, kifestő stb. vezérlő elemek). A legtöbb vezérlő elem ablakban jelenik meg, és futási idő alatt aktivizálható („fogadni tudja a fókuszt"), ez azonban nem követelmény. Természetesen a vezérlő elemek tulajdonságait is változtatni lehet futási időben is, és például tetszés szerint állítgathatjuk vizuális jellemzőit (mekkora legyen a szélessége, magassága, fent, lent, jobbra vagy balra helyezkedjék-e el, sőt az is, hogy látható legyen-e, vagy el legyen rejtve).

Summázat

Mindezen hiányosságok ellenére igen hasznos információs anyag az, amihez az olvasó az ismertetett könyvből hozzájuthat, a Delphi megismeréséhez azonban nem árt más forrásokból is meríteni. Különösen jól jönne, ha magyar nyelven elérhető lenne egy tartalmas ismertetés a Delphi kiegészítéseként megjelent, „Rapid Application Development" (RAD) néven forgalmazott CD-ről, valamint a cég „Component Writer's Guide" című kiadványáról, amelyre a könyv is számos helyen hivatkozik. A RAD a vizuális programfejlesztéshez olyan kincseket tartalmaz, mint egy csomó VBX formátumú komponens, az erőforrás-szerkesztő (Resource WorkShop) és a windowsos Turbo Debugger legújabb verziója, és ami a legfontosabb, rengeteg Delphi komponens forráskódja.

A könyv újabb kiadásához (amely a Delphi elterjedésével hamarosan szükségessé válhat) a kiadó egy lemezmellékletet is csatolhatna. Ezzel ellensúlyozni lehetne az olvasó hiányérzetét, ami többek között abból fakad, hogy a könyvben közölt apró programrészletek nem adnak teljes képet a Delphi gazdag lehetőségeiről. A forrásanyag ehhez is készen áll a szoftver bőséges dokumentációjában és példatárában. Aligha képzelhető el, hogy a szoftverkészítő cég megtagadná a hozzájárulását egy reprezentatív válogatás elkészítéséhez.

Némi bővítés is elkelne az új kiadásban, legalább függelékként hozzáépítve a könyv anyagához. Bár lehet, hogy erre már azért nem nyílik mód, mert időközben piacra került a Delphi 3.0-s változata. A magyar olvasó számára például rendkívül hasznos lenne az ékezetes betűk problémájának korrekt kifejtése: hol, milyen korlátozásokkal kell számolnunk, más korlátokat esetleg hogyan lehet átlépni.

Kondorosi Károly — László Zoltán — Szirmay-Kalos László:

Objektum-orientált szoftverfejlesztés

Első kiadás
ComputerBooks, 1997
430 oldal, 2616 Ft
(lemezmelléklettel).

A másik könyv, amelyet olvasóink figyelmébe ajánlunk, az objektumorientált rendszertervezés és szoftverfejlesztés korszerű, egységes szemléletű kifejtését adja. Ez a jól átgondolt ismeretanyag most jelenik meg először könyv formájában, de elképzelhető, hogy egyes részeit jó néhány olvasónk első kézből ismeri maguktól a szerzőktől, akik az „Objektumorientált programozás", a „Szoftvertechnológia" és az „Objektumorientált szoftverfejlesztés" tantárgyakat oktatják az Informatikai és Villamosmérnöki karok hallgatóinak a Budapesti Műszaki Egyetemen.

Egységes szemléletet!

Figyelemre méltó először is magának a könyvnek a koncepciója. Egy módszertani újítás gyakorlati megvalósítását tűzték ki célul a szerzők: megpróbálják összekapcsolni az objektumorientált modellezés elméletét az implementáció gyakorlatával. Az oktatói tapasztalatok azt mutatják, hogy nagyon nehezen hatol be a programozás gyakorlatába az a sokak által hirdetett nézet, miszerint nem szabad mereven elkülöníteni egymástól az analízis és a tervezés különböző fázisait, és a programfejlesztés egész folyamatában szükség van az egységes szemlélet meghonosítására. A szerzők most arra tesznek kísérletet, hogy végigkísérjék a szoftverfejlesztés minden fázisát, a probléma felvetődésétől és megértésétől kezdve az implementáció részletes kidolgozásáig, bemutatva, hogy a C++ eszközei hogyan használhatók fel e cél érdekében.

Szoftverkatalógust

Igen nagy súlyt fektetnek a szerzők a kifejlesztett programkomponensek újrafelhasználhatóságának biztosítására. Ennek alapvető feltétele, hogy a szoftverfejlesztésben is pontosan definiálva legyenek a több helyen alkalmazható komponensek, továbbá ezek paraméterei, alkalmazási feltételei, és az ezekre vonatkozó ismeretek rendszerezett formában mindenki számára hozzáférhetőek legyenek.

Ahogy a hardverfejlesztők számára készülő katalógusokból könnyen ki lehet választani az adott célnak legjobban megfelelő integrált egységeket, úgy kellene a szoftverfejlesztőknek is hasonló katalógusokat készíteni, és persze elérhetővé és könnyen alkalmazhatóvá tenni magukat a szoftverfejlesztéshez használható komplett részegységeket is.

E cél eléréséhez kiváló lehetőséget biztosít az objektumorientált megközelítés, sajnos azonban sok esetben ez inkább csak jelszó maradt. Gyakran sem a fogalmak, sem a nyelvezet nem tisztázódott eléggé, ahány szerző, annyiféle értelemben használta ugyanazokat az elnevezéseket. Az egész modellezési, tervezési és implementációs folyamat következetes végigkísérése jó lehetőséget ad a szerzőknek az ajánlott módszerek élő működésének a bemutatására. Igen jól segítheti bonyolultabb rendszerek tervezését az objektumokkal való modellezés.



Dinamikusan nyújtózkodó tömbök

A könyv rendkívül érdekes és frappáns anyagából csak egy jellegzetes és részletesen tárgyalt részt szeretnék kiragadni: a tömbök kezelésének problémáját.

A programozói gyakorlatban gyakran okoz fejtörést a tömbök kezelése. Milyen méretűre vegyem? Nem nagy helypazarlás-e mindig a maximális méretekre felkészülni? És egyáltalán, lehet-e tudni, hogy mekkorára dagadhat egy sztring?

A probléma elegáns megoldását mutatják be a szerzők a „dinamikusan nyújtózkodó tömbök" példáján. A példa nem triviális, hiszen a dinamikus adatkezelés megoldása az objektumorientált programozásban sem tartozik a könnyű feladatok közé. Megéri viszont a fáradságot, hogy részproblémák taglalása helyett eléggé általános formában gondoljuk végig az egész kérdéskört. Ezen keresztül ugyanis éppen az objektum-orientált programozás legfőbb előnye válik közvetlenül érzékelhetővé: az újrahasznosíthatóság.

Előfordulhat, hogy egy konkrét alkalmazásban nincs szükségünk egy mindenre kiterjedő általános modell által nyújtott összes lehetőségre. Gondoljuk meg azonban, hogy az így elkészített osztályokból az öröklődés lehetőségeinek kihasználásával játszva készíthetők testre szabott új „mutációk". Megéri!

Aki meg akar ismerkedni a dinamikus tömbök kezelésével és sokoldalú felhasználási lehetőségeivel, annak ajánlom, hogy rágja végig magát ezen a könyvön: sok érdekes és hasznos gondolatot fog találni benne.

Helyenként nehéz követni az árnyaltabb okfejtéseket, de kimondott szellemi élvezetet jelent maga a könyv olvasása is, a hasznosságáról nem is szólva. Sokat segíthetnek a megértéshez (sőt megértésünk teszteléséhez is) a lemezmellékleten található mintafeladatok és a teljes feladatmegoldások.

Vargha Dénes



INFORMÁCIÓKÉRÉS: 0928 ▶



INFORMÁCIÓKÉRÉS: 0927 ▶



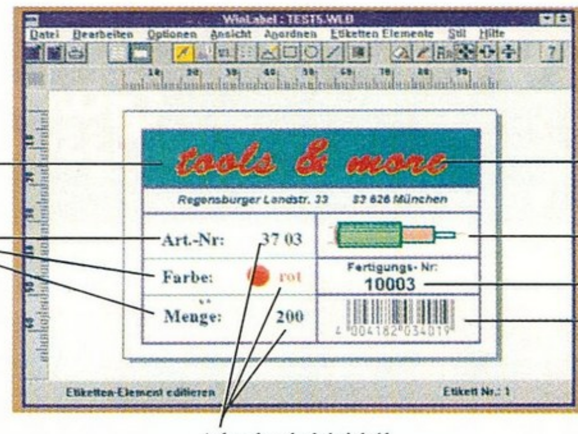
INFORMÁCIÓKÉRÉS: 0930 ▶



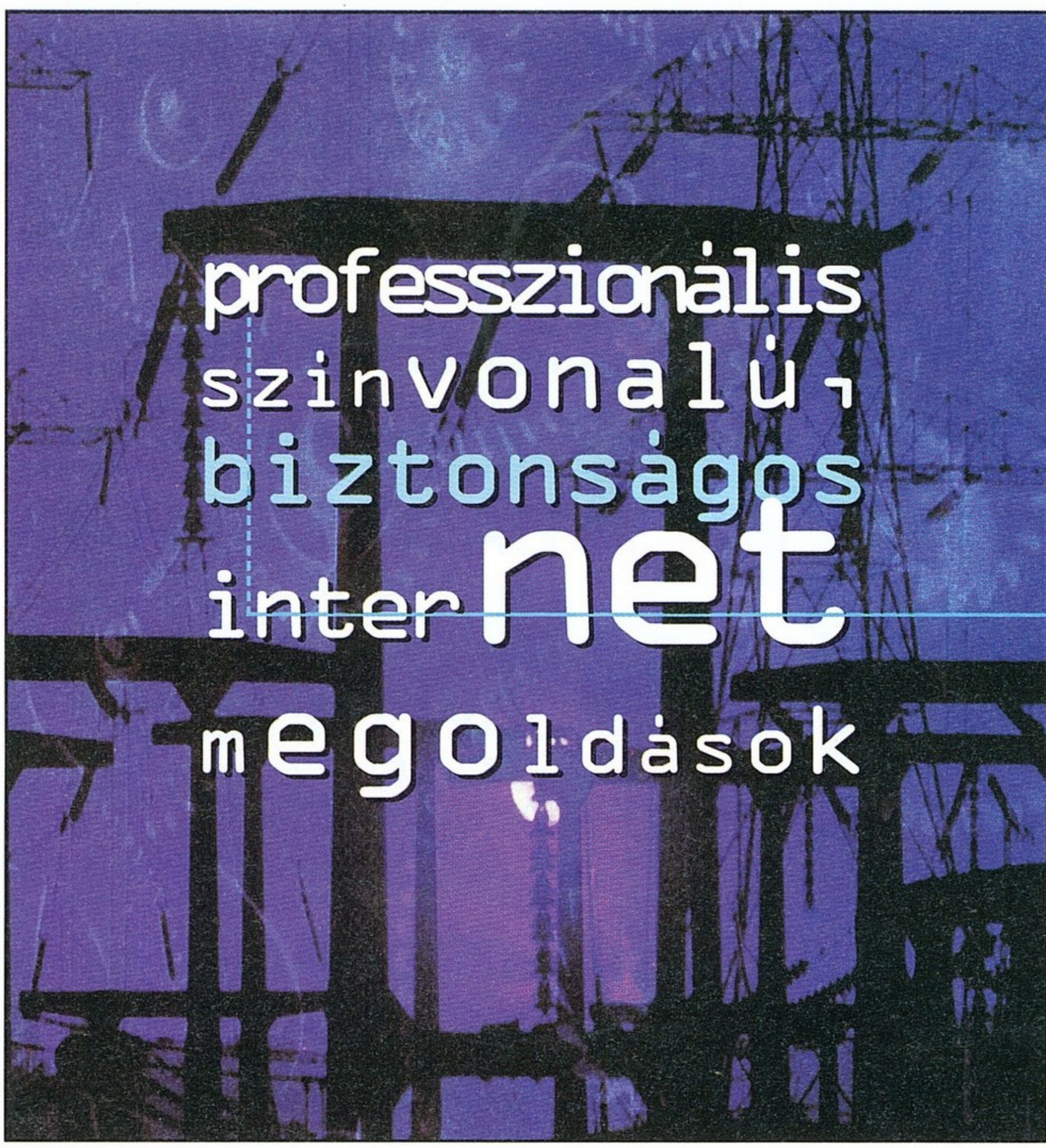
INFORMÁCIÓKÉRÉS: 0942 ▶



INFORMÁCIÓKÉRÉS: 0938 ▶



Háttérszínek
Állandó adatok etikettenként
Különböző fontok és betű méretek
Képek importálása
Sorszámozás
Vonalkódok
Adatok adatbázisból



professzionális
színvonalú,
biztonságos
internet
megoldások

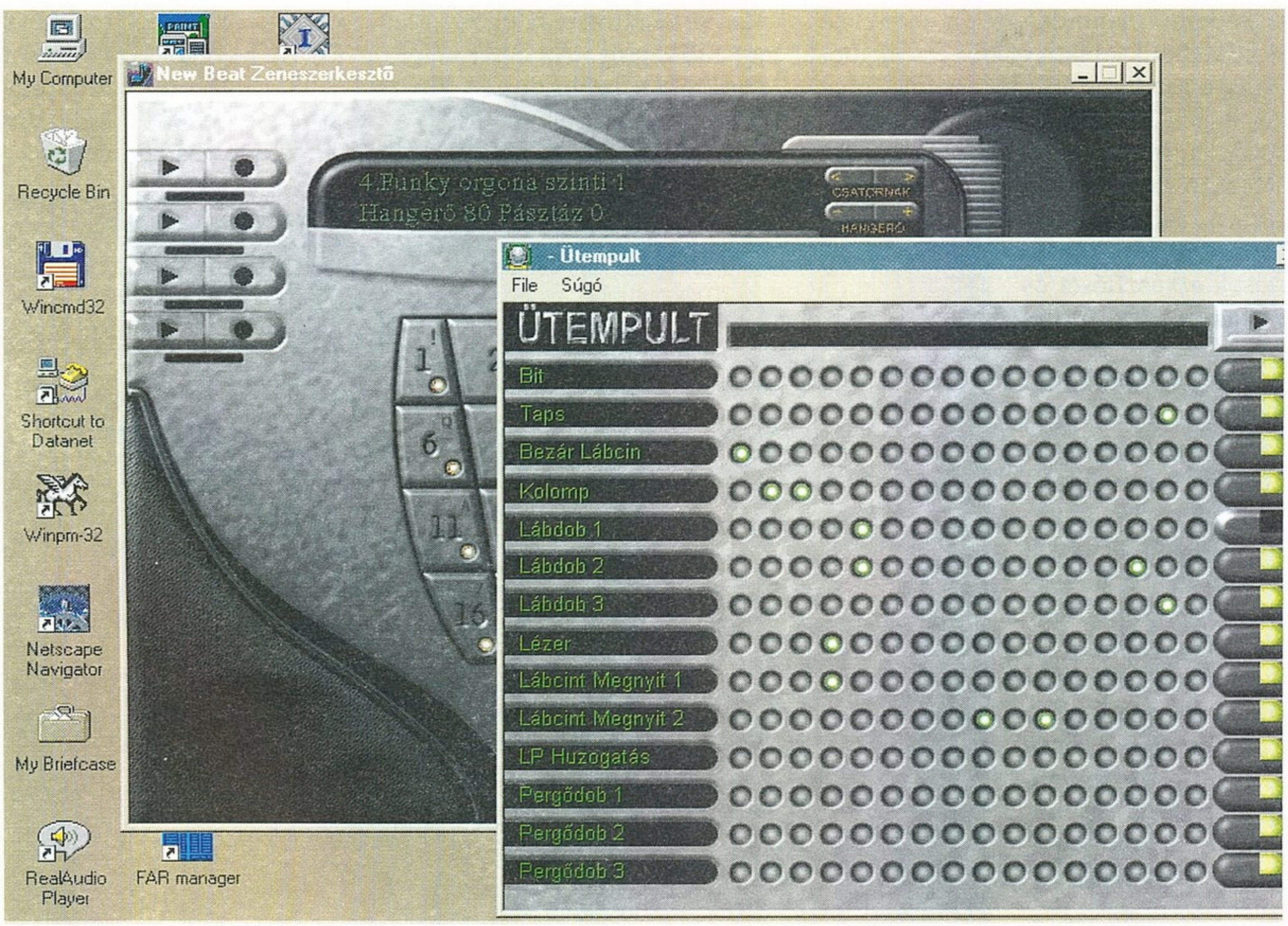
Új zenei révület

A zenekedvelők számára ismerős fogalom a house, a trance, a techno vagy hip-hop, de csak kevesen vannak tisztában azzal, milyen elemekből is építkeznek ezek a műfajok, és hogyan lesz mindezekből az értő fül számára is kellemes hangzás. A Cyberstone kiadásában piacra került NewBeat zeneszerkesztő program a fenti ismeretek gyakorlati megközelítésén túl önálló zenei alkotások készítésének ígéretével is kecsegtet — mégpedig mindenféle zenei előképzettség nélkül.

A profik nyilván fitymálva nyilatkoznak egy ilyen CD-ről, pedig a gyakorlati próba elég meggyőzőnek bizonyult. Igen tiszták például azok a zenei minták, amelyek „helyből" járnak a CD-vel (mintegy 270-et sikerült összeszámlálni a basszus-, dob-, ritmus-, SFX- és vegyes elemek köréből), s ezek puszta összeépítésének variációi révén is hallatlanul gazdag ritmus- és hangzásvilághoz juthat az alkalmi komponista. Arról már nem is beszélve, hogy saját maga alkotta elemekkel tovább bővítheti a választékot (erre szolgál többek között a Sound Warp zeneszerkesztő).

Egy nem elhanyagolható apróság: magának a programnak a felülete úgy van felépítve, hogy egérrel is gyorsan beilleszthető legyen a megfelelő hangelem vagy hangsor, ugyanakkor a „zongoristák" is megkapják a vezérlés lehetőségét a billentyűzeten keresztül. A NewBeat zeneszerkesztő Windows 95 környezetben fut — sajátos felülete miatt azonban nem mindenben igazodik a Win95-ben megszokott rutinokhoz. Ugyanakkor kulturált, könnyen kiismerhető a felépítése, könnyű benne zenét komponálni — és alkotásunkat nehéz elrontani.

Egyáltalán nem túlzás a kiadónak a CD-t kísérő beharangozója: az abszolút kezdőknek is olyan eszköz kerül a kezükbe a NewBeat-tel, amellyel stúdió minőségű zenét szerezhetnek maguknak. A sok pozitívum közepette nem szabad elhallgatni, hogy az eredeti Microforum CD magyarításakor hibák is belekerültek a rendszerbe — például nem onnan nyílik meg egy help-fájl, ahonnan kellene, egyes utasítások magyarázata nem fért bele a nekik szánt szövegmezőbe —, de ezek nem korlátozzák az alkalmi komponisták munkáját vagy önfeledt szórakozását.



Világ telefonjai, egyesüljetek!

Július 24-én, Magyarországnak az Inmarsat nemzetközi mobil űrtávközlési szervezetbe való belépése napján sajtótájékoztatón mutatták be a legújabb Inmarsat mobil terminált, a Mini-M-et. A Nera Telecommunications cég WorldPhone Office készüléke egy önálló, könnyű, hordozható Inmarsat Mini-M terminál, amely hang-, fax-, e-mail- és adatkommunikációt tesz lehetővé a nemzetközi nyilvános telefonhálózaton keresztül. A notebook méretű terminál ideális olyan utazó üzletemberek számára, akik a világon bárhol és bármikor azonnali, kiváló minőségű telefonösszeköttetést igényelnek, és kézenfekvő alkalmazási lehetőség a nemzetközi szállítmányozó cégek fuvarkövető rendszereibe való integráció is. A Nera Mini-M terminált olyan könnyű használni, mint egy hagyományos telefont, kezelése semmilyen technikai ismeretet nem igényel. A WorldPhone tömege 2,4 kg, vékony, kompakt, és pillanatok alatt üzembe helyezhető. A terminál akkumulátora 24 óra készenléti és 2,5 óra beszélgetési időt tesz lehetővé. A forgalmazott műholdas terminálokkal kitűnő minőségű adat- és hangkommunikációs kapcsolat teremthető a nyilvános telefon- illetve adathálózat és a terminálok között, bárhol is legyenek azok a világon. A berendezések átviteli sebessége 600 bit/s és 64 Kbit/s közötti. További jellemzők: 4,8 bit/s hang; Hayes-kompatibilis modem; további telefon- és faxcsatlakozási lehetőség; az akkumulátor töltője a telefonba integrálható; vízálló telefonkészülék; SIM-kártyás felhasználóazonosítás; felhasználóbarát, háttér-világítású LCD kijelző; kézhasználat nélküli üzemmód; többnyelvű menürendszer; beépített telefonkönyv; akusztikus/vizuális jelerősség-indikátor.

A magyarországi Guards Kft. már meg is kezdte a Mini-M terminálok forgalmazását, az árak természetesen még igen magasak.

Oracle8 Alphán és Priorison

Az új Oracle8-as univerzális adatbázis-szerver bemutatásával lényegében egyidejűleg jelentette be a Digital az új termék támogatását és azonnali elérhetőségét az AlphaServeren és x86-alapú Prioris rendszerein. Az Oracle használta fel először a Digital 64 bites környezetét és Unix TruCluster platformját, most pedig a Digital az Oracle8 szerver első platformszállítója, amely kiaknázza a rendszer 64 bites tulajdonságait. A két cég együttműködése más területeken is eredményes: bemutatták például a Digital AlphaServerén futó Oracle videoszervert (multimédia a telekommunikációs, Internet- és intranet-piac számára), valamint a Digital Network Appliance Reference Design alapján készült hálózati számítógép prototípusait, amelyeket StrongARM mikroprocesszorral szereltek, és az Oracle leányvállalatának, a Network Computer Inc. széleskörű szoftverlehetőségeivel ruháztak fel.

Az integrált kereskedelem kulcsa

Augusztus elején mutatta be a Pandesic — az Intel és az SAP közös tulajdonában lévő, kereskedelmi megoldások kifejlesztésére szakosodott vállalat — a Pandesic Internet nevű alkalmazást, amely a szakma első integrált üzleti rendszere. Az új rendszer révén a kereskedők gyorsan és könnyedén szervezhetik meg a teljes folyamatot, amely biztosítja az áruk és szolgáltatások Interneten keresztüli forgalmazását, s jelentősen csökkenthető a Web-hálózatra épülő vállalkozások piaci bevezetéséhez szükséges idő is.

A Pandesic Web kereskedelmi megoldás révén a kereskedők „real time" módon követhetik elektronikus értékesítési folyamataikat. Lévén, hogy a rendszer átfogja a teljes folyamatot, zökkenőmentesen képes kezelni elektronikus kereskedelmi tranzakciók során a marketinget, a megrendelés feldolgozását és teljesítését, a készlet árazását, az anyaggazdálkodást, az adózást, a kifizetések továbbítását, a szállítási kérdéseket, a pénzügyi jelentések elkészítését és a szállítói kifizetésekhez kapcsolódó folyamatokat.

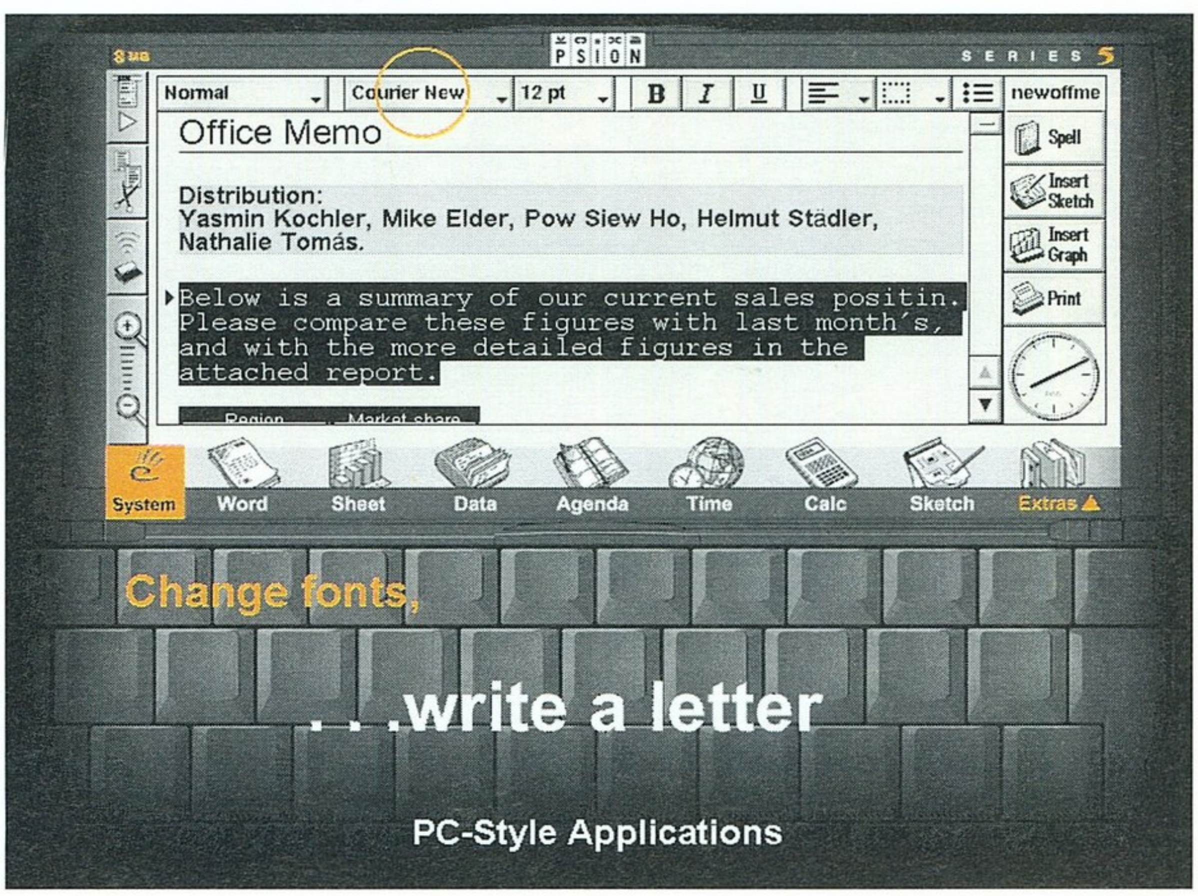
A Pandesic Web kereskedelmi alkalmazást különböző szakterületek számos vállalata támogatja, ide értve a szolgáltatókat, a Web-fejlesztőket, a szoftver- és hardverszállítókat, a pénzügyi intézményeket, az átutalásokkal és logisztikai szolgáltatásokkal foglalkozó cégeket. E cégek között megtalálható a Citibank, a Compaq, a CyberCash, a HP, a Telxon és a US Web Corp. A Pandesic online használatával a kereskedők élvezhetik mindazokat az előnyöket, amelyek a nagyszámú tranzakció esetén modulárisan bővíthető építkezés, valamint a funkcionalitás révén nyílnak meg számukra, így növelhetik Web-alapú vállalkozásaik hatékonyságát, megvethetik lábukat az elektronikus kereskedelem piacán.

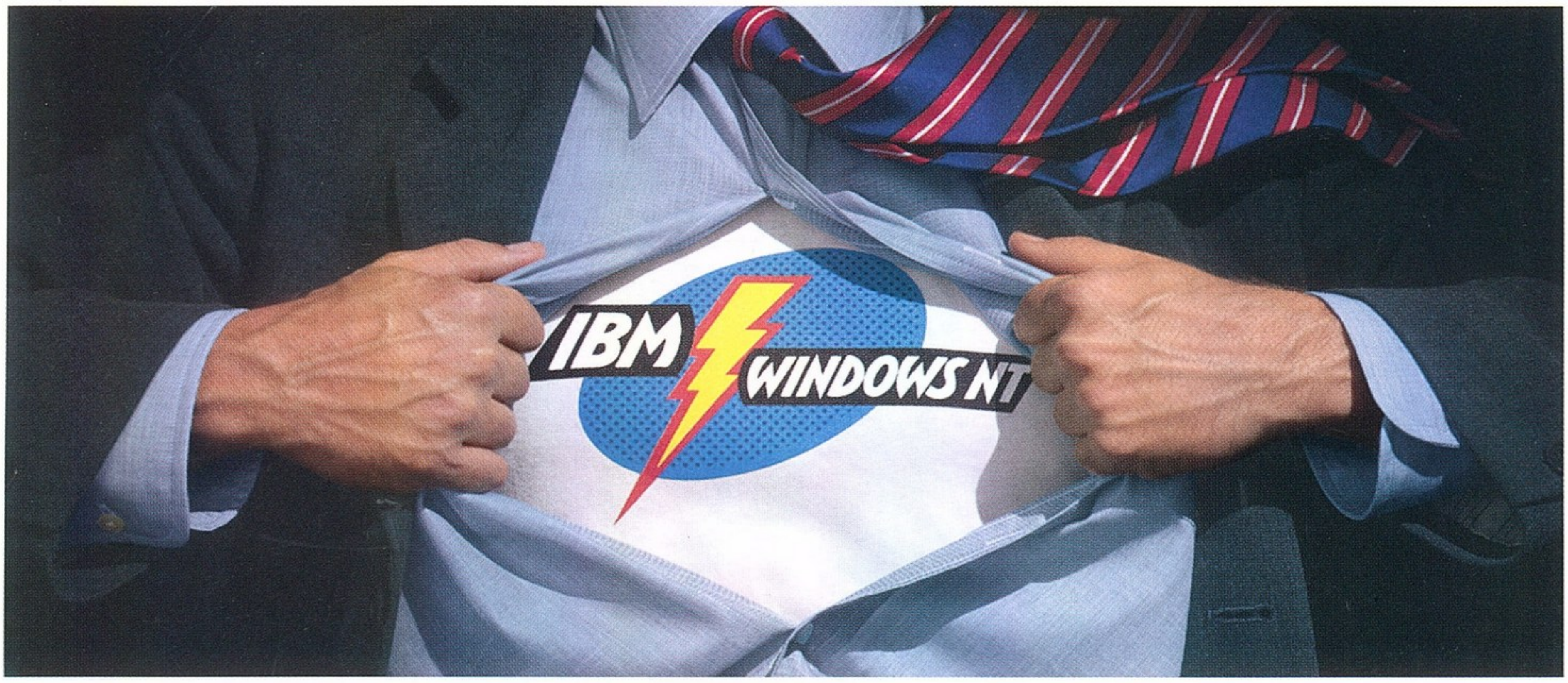
Fogyassz kevesebbet!

Az adatok biztonságának egyik gyenge pontja gyakran a tápegység. Mi történik áramkiesés esetén, mi van, ha lemerül az akkumulátor, kifogy az elem? A legjobb megoldás olyan egységek használata, amelyek az áramkiesés pillanatában még rögzítik az aktuális állapotot, és ha az áramszolgáltatás helyreáll, mintha mi sem történt volna, minden onnan folytatódik tovább. Igen ám, csakhogy ez nagyon költséges. Ezért alkalmazzák a szünetmentes tápokat, a tartós akkumulátorokat, a redundáns tápegységeket.

A kéziszámítógépeknél azonban nyilvánvalóan nincs lehetőség táskanagyságú akkumulátor beépítésére, pláne két példányban. A hosszú élettartamú elemek használata mellett arról az oldalról igyekeztek megközelíteni a problémát, hogy alacsony fogyasztású egységeket helyezzenek üzembe. A belső memóriánál, ugyanúgy, mint a PC-knél, RAM-mal és ROM-mal is dolgoznak, viszont a lemezes adattárolás nem terjedt el. A bővítésekhez kizárólag mozgó alkatrész nélküli elektronikus megoldást alkalmaznak, PCMCIA-t és SSD-t, amelyek az alacsony fogyasztáson túl kisebbek, és mágneses, mechanikus hőhatásokkal szemben ellenállóbbak, ezáltal is növelve az adatbiztonságot. Az SSD fogyasztása csupán harmincadannyi, mint amennyit hagyományos PCMCIA fogyaszt, emellett kapcsolódási felülete is jóval kevesebb érintkezési pontot tartalmaz.

A világpiac egyharmadát uraló Psion például eddig az SSD-t használta, legújabb családtagja, a Series 5 viszont már a most terjedőben levő V. típusú PCMCIA-t, amelynek fent említett paraméterei megközelítik az SSD-éit (például mérete fele a régինek). Várhatóan más palmtopok is áttérnek majd a PCMCIA-slágerre. Mindkét kártya kapacitása 128 KB-tól 16 MB-ig terjed. A V. PCMCIA fejlesztői azt ígérik, hogy egy-két éven belül előrukkolnak több száz MB-os kártyákkal, MB-onként nagyjából tizedáron, azaz egy dollárért. DOS-kompatibilis operációs rendszereknél a tárolt információk szervezésekor részleges hibák esetén is elveszhet az egész adatállomány. Az SSD-nél és a PCMCIA-nál nagy gondot fordítottak robusztus és hibatűrő adatszerkezetek létrehozására, ami azt jelenti, hogy az esetleges hibákat lokalizálni lehet, és így azok nem terjednek ki az egész adathalmazra.





Hogy mennyire veszi komolyan az IBM a Windows NT-t?

Bátran mondhatjuk, hogy manapság senki, még a Microsoft sem fejleszt és integrál erre a platformra annyit, mint az IBM.

Az utóbbi hat hónapban nyolc kulcsfontosságú szoftvert optimalizáltunk a Windows NT részére.

Az első multimédia adatbázistól, a vadonatúj tranzakció szerveren át, a teljes rendszermenedzsmentet biztosító megoldásig segítjük ügyfeleinket a vállalati informatikai eszközök kihasználásában, Windows-os asztali gépeken, vagy bármilyen más platformon behálózott világunkban.

Az egész csomagot, beleértve az ingyenes próbaverziókat, megtalálhatja a www.software.ibm.com/nt honlapon. Addig is még három fontos szó:

skálázhatóság, integráltság, támogatás.

Most már tudja, hol találja őket.

IBM

Nagy megoldások egy kis bolygónak

Amennyiben több információt szeretne kapni, kérjük érdeklődjön telefonon, vagy a kupont kitöltve faxolja, illetve küldje vissza címünkre!

Cégnév: __________

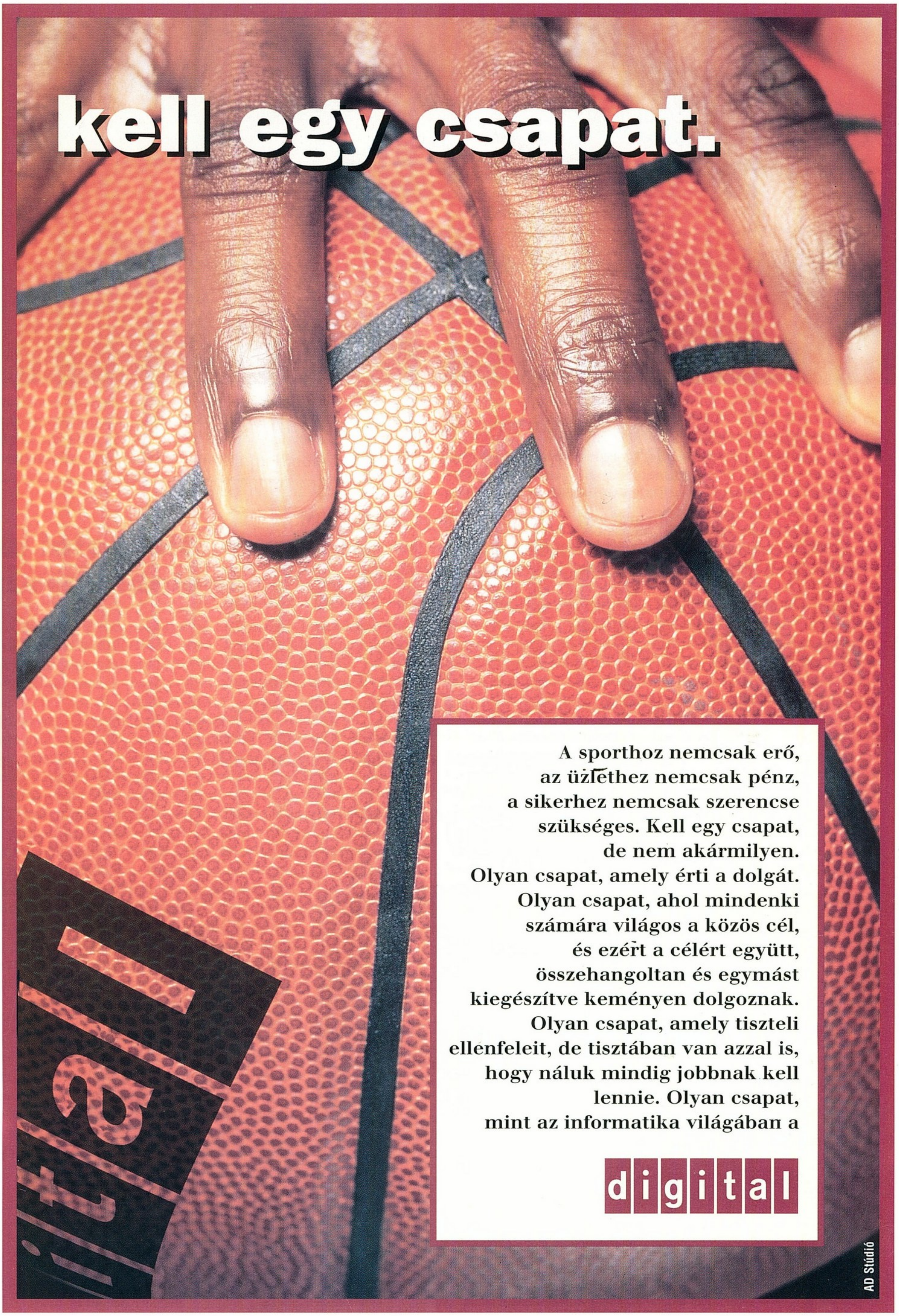
Név: __________

Telefon: __________

Cím: IBM Direkt Marketing, 1119 Budapest, Keveháza u. 1.
tel.: 204-1981, fax: 204-1530

IBM

INFORMÁCIÓKÉRÉS: 0914 ▲



INFORMÁCIÓKÉRÉS: 0901 ▶